BỘ GIÁO DỤC VÀ ĐÀO TẠO

CHUYÊN ĐỀ HỌC TẬP TIN HỌC 11

ĐỊNH HƯỚNG KHOA HỌC MÁY TÍNH

SÁCH GIÁO VIÊN



KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM

PHẠM THẾ LONG (Tổng Chủ biên)
BÙI VIỆT HÀ (Chủ biên)
NGUYỄN HOÀNG HÀ – NGUYỄN THỊ HIỀN
TRƯƠNG VÕ HỮU THIÊN – LÊ HỮU TÔN – PHẠM THỊ BÍCH VÂN

CHUYÊN ĐỀ HỌC TẬP TIN HỌC 11

ĐỊNH HƯỚNG KHOA HỌC MÁY TÍNH

SÁCH GIÁO VIÊN

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM

QUY ƯỚC VIẾT TẮT DÙNG TRONG SÁCH

CS	Khoa học máy tính
GDPT	giáo dục phổ thông
GV	giáo viên
HS	học sinh
ICT	Tin học ứng dụng
PPDH	phương pháp dạy học
SCĐ	sách chuyên đề
SGK	sách giáo khoa
SGV	sách giáo viên
THCS	Trung học cơ sở
THPT	Trung học phổ thông

KẾT NỐI TRI THỨC
VỚI CUỘC SỐNG

LỜI NÓI ĐẦU

Các thầy giáo, cô giáo thân mến!

Trên tay các thầy cô là cuốn *Chuyên đề học tập Tin học 11 – Định hướng Khoa học máy tính*. Trong Chương trình môn Tin học của Chương trình giáo dục phổ thông 2018 do Bộ Giáo dục và Đào tạo ban hành theo Thông tư 32/2018/TT–BGDĐT, thời lượng dành cho Chuyên đề học tập là 35 tiết.

Đây là cuốn sách chuyên đề rất đặc biệt, lần đầu tiên, một số kiến thức cơ bản của lí thuyết và kĩ thuật thiết kế thuật toán được đưa vào nhà trường phổ thông. Nội dung của chuyên đề này được thiết kế độc lập với chương trình chính khoá, nhưng học sinh cần biết một số kiến thức ban đầu của phần kĩ thuật lập trình trong chương trình sách giáo khoa Tin học 11 trước khi có thể nắm vững được kiến thức của chuyên đề này.

Sách *Chuyên đề học tập Tin học 11 – Định hướng Khoa học máy tính* gồm 3 chuyên đề thành phần, mỗi chuyên đề thành phần là kiến thức của một kĩ năng thiết kế thuật toán rất quan trọng mà mỗi người làm khoa học máy tính cần nắm vững.

– Chuyên đề 1 trình bày kĩ thuật đầu tiên và cơ bản của lập trình.

– Chuyên đề 2 trình bày về phương pháp và kĩ thuật chia để trị, một trong những phương pháp thiết kế thuật toán kinh điển.

– Chuyên đề 3 trình bày về kĩ thuật duyệt vét cạn, một kĩ thuật duyệt tương đối khó của thiết kế thuật toán.

Chuyên đề học tập Tin học 11 – Định hướng Khoa học máy tính là nội dung mới, tương đối khó và là cơ sở nền móng của khoa học máy tính. Các tác giả mong muốn nhận được nhiều góp ý của các thầy cô giáo để cuốn sách được hoàn thiện hơn.

Chúc các thầy cô giáo thành công!

Các tác giả

MỤC LỤC

Trang

PHẦN MỘT. HƯỚNG DẪN CHUNG 5

I. Mục tiêu môn học 5

II. Giới thiệu sách "Chuyên đề học tập Tin học 11 – Định hướng Khoa học máy tính" ... 8

III. Phương pháp dạy học. 10

IV. Kiểm tra, đánh giá. 12

PHẦN HAI. HƯỚNG DẪN DẠY HỌC CÁC BÀI CỤ THỂ 14

Chuyên đề 1. Thực hành thiết kế thuật toán theo kĩ thuật đệ quy 16

Bài 1. Đệ quy và hàm đệ quy 16

Bài 2. Thiết kế thuật toán bằng đệ quy 19

Bài 3. Thực hành giải toán theo kĩ thuật đệ quy 23

Bài 4. Bài toán Tháp Hà Nội 26

Bài 5. Thực hành thiết kế thuật toán theo kĩ thuật đệ quy 30

Chuyên đề 2. Thực hành thiết kế thuật toán theo kĩ thuật chia để trị 33

Bài 6. Ý tưởng và kĩ thuật chia để trị 33

Bài 7. Thiết kế thuật toán theo kĩ thuật chia để trị 38

Bài 8. Thực hành thiết kế thuật toán tìm kiếm theo kĩ thuật chia để trị 42

Bài 9. Sắp xếp trộn 46

Bài 10. Thực hành giải toán bằng kĩ thuật chia để trị 54

Chuyên đề 3. Thực hành thiết kế thuật toán theo kĩ thuật duyệt 59

Bài 11. Bài toán tìm kiếm và kĩ thuật duyệt 59

Bài 12. Thực hành kĩ thuật duyệt cho bài toán tìm kiếm 64

Bài 13. Kĩ thuật duyệt quay lui 67

Bài 14. Thực hành kĩ thuật duyệt quay lui 72

Bài 15. Bài toán xếp Hậu 75

Bài 16. Thực hành thiết kế thuật toán theo kĩ thuật duyệt quay lui 78

PHẦN
MỘT HƯỚNG DẪN CHUNG

Chương trình môn học nằm trong phần các chuyên đề của môn Tin học 11, định hướng Khoa học máy tính. Trong Chương trình giáo dục phổ thông 2018, mỗi HS cấp THPT bắt buộc phải chọn học 3 chuyên đề học tập với tổng thời lượng 105 tiết. Như vậy, mỗi HS chọn học Tin học đều có quyền chọn môn chuyên đề Tin học định hướng Khoa học máy tính để học.

I. MỤC TIÊU MÔN HỌC

1. MỤC TIÊU CHUNG

Như chúng ta đã biết, trong chương trình tổng thể môn Tin học trong nhà trường phổ thông đã đưa ra mục tiêu tổng quát là:

Chương trình môn Tin học ở cấp THPT giúp HS củng cố và nâng cao năng lực tin học đã được hình thành, phát triển ở giai đoạn giáo dục cơ bản, đồng thời cung cấp cho HS tri thức mang tính định hướng nghề nghiệp thuộc lĩnh vực tin học hoặc ứng dụng tin học. Cụ thể nhằm:

– Giúp HS có những hiểu biết cơ bản về hệ thống máy tính, một số kĩ thuật thiết kế thuật toán, tổ chức dữ liệu và lập trình; củng cố và phát triển hơn nữa cho HS tư duy giải quyết vấn đề, khả năng đưa ra ý tưởng và chuyển giao nhiệm vụ cho máy tính thực hiện.

– Giúp HS có khả năng ứng dụng tin học, tạo ra sản phẩm số phục vụ cộng đồng và nâng cao hiệu quả công việc; có khả năng lựa chọn, sử dụng, kết nối các thiết bị số, dịch vụ mạng và truyền thông, phần mềm và các tài nguyên số khác.

– Giúp HS có khả năng hoà nhập và thích ứng được với sự phát triển của xã hội số, ứng dụng công nghệ thông tin và truyền thông trong học và tự học; tìm kiếm và trao đổi thông tin theo cách phù hợp, tuân thủ pháp luật, có đạo đức, ứng xử văn hoá và có trách nhiệm; có hiểu biết thêm một số ngành nghề thuộc lĩnh vực tin học, chủ động và tự tin trong việc định hướng nghề nghiệp tương lai của bản thân.

Với các mục tiêu trên, rõ ràng môn Tin học sẽ có ba mạch kiến thức, định hướng chính là:

Học vấn số hoá phổ thông (DL) nhằm giúp HS có khả năng hoà nhập với xã hội hiện đại, sử dụng được các thiết bị số và phần mềm cơ bản thông dụng một cách có đạo đức, văn hoá và tuân thủ pháp luật.

Công nghệ thông tin và truyền thông (ICT) nhằm giúp HS có khả năng sử dụng và áp dụng hệ thống máy tính giải quyết vấn đề thực tế một cách hiệu quả và sáng tạo.

Khoa học máy tính (CS) nhằm giúp HS hiểu biết các nguyên tắc cơ bản và thực tiễn của tư duy máy tính; tạo cơ sở cho việc thiết kế và phát triển các hệ thống máy tính.

Môn Tin học hình thành và phát triển ở HS năm năng lực thành phần của năng lực tin học:

– NLa: Sử dụng và quản lí các phương tiện công nghệ thông tin và truyền thông;

– NLb: Ứng xử phù hợp trong môi trường số;

– NLc: Giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông;

– NLd: Ứng dụng công nghệ thông tin và truyền thông trong học và tự học;

– NLe: Hợp tác trong môi trường số.

2. MỤC TIÊU CẤP HỌC

Chương trình môn Tin học ở cấp THPT giúp HS củng cố và nâng cao năng lực tin học đã được hình thành, phát triển ở giai đoạn giáo dục cơ bản, đồng thời cung cấp cho HS tri thức mang tính định hướng nghề nghiệp thuộc lĩnh vực khoa học máy tính hoặc ứng dụng tin học, cụ thể là:

– Giúp HS có những hiểu biết cơ bản về hệ thống máy tính, một số kĩ thuật thiết kế thuật toán, tổ chức dữ liệu và lập trình; củng cố và phát triển hơn nữa cho HS tư duy giải quyết vấn đề, khả năng đưa ra ý tưởng và chuyển giao nhiệm vụ cho máy tính thực hiện.

– Giúp HS có khả năng ứng dụng tin học, tạo ra sản phẩm số phục vụ cộng đồng và nâng cao hiệu quả công việc; có khả năng lựa chọn, sử dụng, kết nối các thiết bị số, dịch vụ mạng và truyền thông, phần mềm và các tài nguyên số khác.

– Giúp HS có khả năng hoà nhập và thích ứng được với sự phát triển của xã hội số, ứng dụng công nghệ thông tin và truyền thông trong học và tự học; tìm kiếm và trao đổi thông tin theo cách phù hợp, tuân thủ pháp luật, có đạo đức, ứng xử văn hoá và có trách nhiệm; có hiểu biết thêm một số ngành nghề thuộc lĩnh vực tin học, chủ động và tự tin trong việc định hướng nghề nghiệp tương lai của bản thân.

3. MỤC TIÊU LỚP

Nội dung chính của sách *Chuyên đề học tập Tin học 11 – Định hướng Khoa học máy tính* gồm ba chuyên đề thành phần như sau:

Yêu cầu cần đạt	Nội dung
Chuyên đề 1: Thực hành thiết kế thuật toán theo kĩ thuật đệ quy (10 tiết)	
– Trình bày được tính đệ quy trong một vài định nghĩa sự vật, sự việc. – Xác định được phần cơ sở và phần đệ quy trong mô tả đệ quy. – Ứng dụng được kĩ thuật *Đệ quy* trong thiết kế một vài thuật toán như: + Tính a^n (hoặc n!). + Tìm phần tử thứ n của dãy Fibonacci. + Bài toán Tháp Hà Nội. – Viết được chương trình sử dụng kĩ thuật *Đệ quy* cho một vài bài toán đơn giản. – Nhận biết được tính ưu việt của kĩ thuật *Đệ quy* trong định nghĩa sự vật, mô tả và thiết kế thuật toán.	– Khái niệm *Đệ quy* – Thực hành thiết kế thuật toán theo kĩ thuật *Đệ quy*
Chuyên đề 2: Thực hành thiết kế thuật toán theo kĩ thuật chia để trị (10 tiết)	
– Giải thích được sơ lược về kĩ thuật *Chia để trị* (Divide and Conquer). – Nêu được ý tưởng thiết kế thuật toán theo kĩ thuật *Chia để trị*, giải thích được mối liên hệ giữa thiết kế thuật toán theo kĩ thuật *Chia để trị* và *Đệ quy*, nêu được ví dụ minh hoạ. – Viết được chương trình đơn giản có sử dụng kĩ thuật *Chia để trị*.	Thực hành thiết kế thuật toán theo kĩ thuật *Chia để trị*
Chuyên đề 3: Thực hành thiết kế thuật toán theo kĩ thuật duyệt (15 tiết)	
– Nêu được ý tưởng của kĩ thuật *Duyệt* và ví dụ minh hoạ (như tìm phần tử lớn nhất hoặc nhỏ nhất trong một dãy số, tìm một số trong một dãy số,...). – Nêu được ý tưởng của kĩ thuật *Quay lui* và nêu được ví dụ minh hoạ (như in các xâu nhị phân độ dài n, tìm tất cả các hoán vị của n phần tử,...). – Nhận ra được mối liên quan giữa thiết kế thuật toán theo kĩ thuật *Quay lui* và kĩ thuật *Đệ quy*. – Viết được chương trình đơn giản có sử dụng kĩ thuật *Duyệt*. – Viết được chương trình đơn giản có sử dụng kĩ thuật *Quay lui*.	– Kĩ thuật *Duyệt* – Kĩ thuật *Quay lui*

II. GIỚI THIỆU SÁCH CHUYÊN ĐỀ HỌC TẬP TIN HỌC 11 – ĐỊNH HƯỚNG KHOA HỌC MÁY TÍNH

1. MỘT SỐ ĐIỂM CẦN CHÚ Ý VỀ NỘI DUNG

Các GV cần hiểu rõ một số điểm khác biệt và quan trọng của nội dung chuyên đề này như sau:

a) Đây là những kiến thức đầu tiên của thiết kế thuật toán theo một số kĩ thuật cụ thể như đệ quy, chia để trị và quay lui. Các kĩ thuật này đều là những kĩ thuật rất cơ bản của lí thuyết thuật toán. Các kiến thức này lần đầu tiên được đưa vào nhà trường phổ thông và chuyên đề này là những bài học đầu tiên. Đây là những kiến thức rất khó với HS. GV cần hiểu rõ các kĩ thuật này để có thể truyền đạt được đầy đủ đến với HS.

b) Về nguyên tắc nội dung của chuyên đề này không có liên quan trực tiếp với phần kĩ thuật lập trình của SGK *Tin học 11 - Định hướng Khoa học máy tính*. Tuy nhiên, giữa hai tài liệu này liên quan nhiều với nhau về chuyên môn, các GV cần nắm rõ để chủ động trong khi giảng dạy. Một số liên quan giữa phần kĩ thuật lập trình của SGK *Tin học 11 - CS* và nội dung sách Chuyên đề CS như sau:

– Thuật toán tìm kiếm nhị phân được học trong phần kĩ thuật lập trình của SGK sẽ được dùng để minh hoạ cho kĩ thuật lập trình đệ quy và kĩ thuật chia để trị.

– Toàn bộ các thuật toán được học trong SCĐ CS này đều là các phương pháp thiết kế cụ thể hơn của phương pháp thiết kế làm mịn dần, hay chính là cách tiếp cận từ trên xuống đã được học trong SGK Tin học 11-CS.

– Trong tất cả các bài học của SCĐ CS, HS mặc định phải biết về độ phức tạp thuật toán và có thể hiểu được các đánh giá độ phức tạp thời gian thuật toán một cách đơn giản.

c) Một điểm nữa cần chú ý với nội dung của chuyên đề: cách biểu diễn thuật toán. Hiện nay trong các tài liệu về khoa học máy tính thường có nhiều cách để mô tả hay biểu diễn thuật toán. Một số cách mô tả thuật toán phổ biến là:

1) Mô tả thuật toán bằng cách liệt kê các bước.

2) Mô tả thuật toán bằng sơ đồ khối.

3) Sử dụng mã giả (pseudocode) để biểu diễn thuật toán. Mã giả thực chất là một "ngôn ngữ lập trình" hoàn chỉnh, nhưng cho phép viết một số lệnh không cần hoàn chỉnh. Mô tả thuật toán bằng mã giả có ưu điểm là không phụ thuộc vào một ngôn ngữ lập trình cụ thể, thường viết mã giả gần với ngôn ngữ tự nhiên nên dễ hiểu hơn. Tuy nhiên, nhược điểm của mã giả là không có sự thống nhất trong cách viết, nên nhiều khi gây khó khăn cho người đọc.

4) Mô tả thuật toán bằng một ngôn ngữ lập trình cụ thể.

Trong SCĐ này đã lựa chọn cách 4 vì ngôn ngữ Python rất đơn giản, dễ hiểu và có cách viết rất gần với ngôn ngữ tự nhiên, thuận tiện để biểu diễn thuật toán.

2. CẤU TRÚC SÁCH

SCĐ gồm ba chuyên đề thành phần:

– Chuyên đề 1 giúp HS tìm hiểu, biết và có kĩ năng thiết kế và giải các bài toán bằng kĩ thuật đệ quy.

– Chuyên đề 2 giúp HS học và tìm hiểu một phương pháp thiết kế rất quan trọng trong lập trình là kĩ thuật chia để trị.

– Chuyên đề 3 giúp HS tìm hiểu và bước đầu nắm được cách giải quyết bài toán theo kĩ thuật duyệt quay lui, một kĩ thuật thường được coi là rất khó trong lí thuyết thuật toán.

Cấu trúc sách *Chuyên đề học tập Tin học 11 – Định hướng Khoa học máy tính* như sau:

STT	Tên chuyên đề	Tên bài học	Số tiết
1	Chuyên đề 1. Thực hành thiết kế thuật toán theo kĩ thuật đệ quy.	Bài 1. Đệ quy và hàm đệ quy	2
		Bài 2. Thiết kế thuật toán bằng đệ quy	2
		Bài 3. Thực hành giải toán theo kĩ thuật đệ quy	2
		Bài 4. Bài toán Tháp Hà Nội	2
		Bài 5. Thực hành thiết kế thuật toán theo kĩ thuật đệ quy	2
2	Chuyên đề 2. Thực hành thiết kế thuật toán theo kĩ thuật chia để trị.	Bài 6. Ý tưởng và kĩ thuật chia để trị	2
		Bài 7. Thiết kế thuật toán theo kĩ thuật chia để trị	2
		Bài 8. Thực hành thiết kế thuật toán tìm kiếm theo kĩ thuật chia để trị	2
		Bài 9. Sắp xếp trộn	2
		Bài 10. Thực hành giải toán bằng kĩ thuật chia để trị	2
3	Chuyên đề 3. Thực hành thiết kế thuật toán theo kĩ thuật duyệt.	Bài 11. Bài toán tìm kiếm và kĩ thuật duyệt	2
		Bài 12. Thực hành kĩ thuật duyệt cho các bài toán tìm kiếm	2
		Bài 13. Kĩ thuật duyệt quay lui	3
		Bài 14. Thực hành kĩ thuật duyệt quay lui	2
		Bài 15. Bài toán xếp Hậu	3
		Bài 16. Thực hành thiết kế thuật toán theo kĩ thuật duyệt quay lui.	3

Các bài học đều được xây dựng với cấu trúc thống nhất, bao gồm những mục chính sau:

– *Mục tiêu bài học* được đặt trong khung với lời dẫn "Sau bài học này em sẽ", tiếp theo là những chỉ báo có thể quan sát được về yêu cầu cần đạt sau bài học.

– *Phần khởi động bài học* định hướng vào vấn đề sẽ được giải quyết trong bài học, được trình bày dưới dạng đoạn văn mô tả hoặc câu hỏi đặt ra những tình huống, gợi mở vấn đề, nhằm thu hút sự chú ý của HS vào nội dung bài học.

– *Phần nội dung bài học* được trình bày cô đọng, kèm theo hình minh hoạ để HS có thể tự học hoặc học tập với sự hướng dẫn của GV. Các hoạt động đặt trong phần nội dung là những gợi ý để tổ chức lớp học tích cực, giúp cho HS chủ động hơn trong quá trình nhận thức.

– *Phần câu hỏi* ngay sau nội dung kiến thức mới nhằm hỗ trợ kiểm tra nhanh kiến thức vừa học.

– *Phần thực hành* gồm các nhiệm vụ được chia nhỏ và hướng dẫn từng bước tư duy, lên kế hoạch để thực hiện nhiệm vụ bằng những công cụ đã được học.

– *Phần luyện tập* gồm những câu hỏi, bài tập nhằm củng cố kiến thức, kĩ năng của bài học cho HS, giúp HS tiếp tục hoàn thiện dự án.

– *Phần vận dụng* gồm những câu hỏi, bài tập nhằm hình thành năng lực vận dụng của HS thông qua sự kết hợp giữa nội dung bài học và kiến thức, kĩ năng đã có từ trước hoặc được hình thành từ thực tiễn cuộc sống để nâng cấp sản phẩm dự án hoặc sáng tạo các sản phẩm số khác.

Cùng với các thành phần trên, những khái niệm/kiến thức mới có trong các bài đều được đúc kết và trình bày ngắn gọn trong các *hộp kiến thức* giúp HS dễ ghi nhớ để vận dụng sau này.

Phần cuối của sách là bảng giải thích các thuật ngữ mới giúp HS dễ dàng tra cứu khi ôn tập cũng như khi tự học, tự rèn luyện kĩ năng, củng cố năng lực.

III. PHƯƠNG PHÁP DẠY HỌC

1. Chương trình phổ thông môn Tin học định hướng phương pháp dạy học và tổ chức hình thức dạy học chung với một số nội dung sau:

a) Áp dụng các phương pháp dạy học tích cực, coi trọng dạy học trực quan và thực hành. Khuyến khích sử dụng phương pháp dạy học theo dự án. Việc dạy học ở phòng thực hành máy tính cần được tổ chức linh hoạt.

b) Tuỳ theo nội dung bài, ở mỗi hoạt động, lựa chọn hình thức tổ chức dạy học phù hợp. Một số chủ đề liên quan trực tiếp đến lập luận, suy diễn lôgic, tư duy thuật toán và giải quyết vấn đề có thể được dạy học không nhất thiết phải sử dụng máy tính.

c) Gắn nội dung kiến thức với các vấn đề thực tế, yêu cầu HS không chỉ đề xuất giải pháp cho vấn đề mà còn phải biết kiểm chứng hiệu quả của giải pháp thông qua sản phẩm số.

d) Chú ý thực hiện dạy học phân hoá. Ở cấp trung học cơ sở, giúp HS lựa chọn những chủ đề thích hợp, khơi gợi niềm đam mê và giúp HS phát hiện khả năng của bản thân đối với môn Tin học, chuẩn bị cho sự lựa chọn môn Tin học ở cấp THPT.

2. Đặc điểm của phương pháp dạy học trong Tin học là bài học được tiến hành dựa trên sự đa dạng của các hình thức tổ chức lớp học. Những phương pháp truyền thống như thuyết trình hay dạy học nêu vấn đề vẫn sẽ được sử dụng trên lớp kết hợp với những phương pháp dạy học tích cực khác, tuỳ theo điều kiện cụ thể.

Một trong những phương pháp dạy học tích cực, phù hợp với Tin học là dạy học dựa trên các hoạt động. Hoạt động được thể hiện trong cấu trúc của mỗi bài và được chỉ dẫn chi tiết trong phần hướng dẫn cụ thể. Tuy nhiên, các hoạt động đều có một số điểm chung là:

a) Khuyến khích HS làm việc cộng tác. HS được chia thành các nhóm theo nhiều cách khác nhau, được rèn luyện kĩ năng nhóm qua các hoạt động.

b) Mọi hoạt động đều có sản phẩm. Việc hoàn thành sản phẩm gắn liền với kĩ năng và kiến thức của mỗi bài học. Trước khi bắt đầu các hoạt động độc lập, cần chuẩn bị vật liệu và thống nhất tiêu chí đánh giá sản phẩm.

c) Mỗi hoạt động đều có ba giai đoạn: 1) Trao đổi toàn lớp để nêu yêu cầu hoạt động và cách đánh giá; 2) Làm việc độc lập hoặc theo các nhóm, để hoàn thành sản phẩm; 3) Toàn lớp tập trung để đánh giá, nhận xét và kết luận.

d) Đa dạng hoá hình thức đánh giá. Mọi hoạt động của HS đều được quan sát, các sản phẩm đều được đánh giá và được ghi chép lại. Đánh giá là kết quả của việc tổng hợp kết quả của các hoạt động thay vì chỉ sử dụng hình thức bài kiểm tra (trắc nghiệm khách quan, tự luận hoặc thực hành).

3. Với SCĐ Tin học 11 CS, GV có thể kết hợp một số phương pháp dạy học như sau:

• Với các tiết học lí thuyết:

Hầu hết các bài lí thuyết của chuyên đề này đều có nội dung thiết kế thuật toán mới. Quy trình thiết lập một thuật toán mới có thể như sau (kết hợp các cách tổ chức dạy học khác nhau trên lớp).

– Giới thiệu bài toán, ghi ra đầu vào, đầu ra chuẩn theo yêu cầu.

– Nêu ý tưởng xuất phát ban đầu để gợi ý thuật toán và cách giải.

– Quan sát và thực hiện thuật toán với các trường hợp đặc biệt, từ đó có thể tổng quát hoá hoặc tìm ra quy luật mở rộng.

– Mô tả thuật toán chính.

– Biểu diễn thuật toán.

– Cài đặt chạy cho một trường hợp với số liệu cụ thể, đơn giản.

– Rút ra kết luận và chốt kiến thức thông qua các câu hỏi củng cố.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

• Với các tiết thực hành:

– GV cần nhắc lại phần lí thuyết của thuật toán, gợi ý cho HS đọc đề bài và thảo luận cách áp dụng thuật toán tương ứng đã học.

– HS tiến hành cài đặt chương trình theo mẫu đã có trong sách.

– HS tiến hành kiểm thử thông qua các bộ dữ liệu kiểm thử (test).

IV. KIỂM TRA, ĐÁNH GIÁ

1. ĐO LƯỜNG VÀ ĐÁNH GIÁ

Đánh giá (Evaluation) là nhận định về mức độ đạt được của HS so với yêu cầu môn học. Đối với môn Tin học, yêu cầu đó được thể hiện dưới dạng những năng lực và được cụ thể hoá thành những yêu cầu trong mỗi bài học.

Đánh giá được thực hiện ba giai đoạn của chu kì dạy học: 1) Đánh giá đầu tiến trình dạy học để chẩn đoán (diagnostic); 2) Đánh giá trong tiến trình (formative) dạy học để thu được thông tin phản hồi nhằm điều chỉnh quá trình dạy học; 3) Đánh giá khi kết thúc tiến trình dạy học (summative) để tổng kết.

Đánh giá sử dụng cả những phương pháp định tính (qualitative) và những phương pháp định lượng (quantitative). Tuy nhiên, cả hai phương pháp đánh giá đều phải dựa trên sự đo lường. Đo lường (measurement) là phương pháp đánh giá định lượng dựa trên số hoặc một thang giá trị nhất định. Thông thường, đo lường trong giáo dục thường được thực hiện qua các bài kiểm tra. Bài kiểm tra có thể được thực hiện theo phương pháp chủ quan (subjective) hay khách quan (objective).

Dựa trên những nguyên tắc đánh giá đó, trong chuyên đề Tin học định hướng CS các GV cần thiết lập các tiêu chí đánh giá như sau:

– Tiêu chí đánh giá cần được bám sát vào mục tiêu bài học. Yêu cầu quan trọng của các tiêu chí là có thể quan sát và đo lường được để có thể quy chiếu vào thang đánh giá.

– Mọi hoạt động của HS đều được quan sát, các sản phẩm đều được đánh giá và được ghi chép lại, nhằm theo dõi sự tiến bộ của các em.

– Đánh giá thường xuyên và chẩn đoán được tổng hợp từ kết quả của các hoạt động đa dạng (kể cả thực hành máy tính) trong mỗi bài học.

– Bài kiểm tra dưới dạng trắc nghiệm khách quan, tự luận hoặc thực hành thường được sử dụng để đánh giá kết thúc học kì.

2. CÁC KĨ THUẬT ĐÁNH GIÁ

– Quan sát HS trong khi thực hiện các hoạt động.

– Đối chiếu sản phẩm sau hoạt động với các tiêu chí đã được thống nhất.

– Thảo luận.

– Trắc nghiệm khách quan.

– Tự luận.

– Nghiên cứu hồ sơ cá nhân của HS.

– HS tự đánh giá và đánh giá lẫn nhau.

– Vấn đáp và câu đố.

– Đánh giá theo nhóm.

– Hỏi và lắng nghe ý kiến của HS.

3. ĐÁNH GIÁ QUA CÁC HOẠT ĐỘNG

– Đặc điểm của các hoạt động có sử dụng công nghệ và các hình thức hợp tác là đánh giá công việc của HS thường mang tính chủ quan. Quan trọng là GV tạo ra một chuẩn để đo tính chủ quan như vậy.

– Sử dụng bảng tương quan giữa nội dung và thang điểm hoặc kí hiệu mức độ (rubric) là một cách tạo ra sự nhất quán và đáng tin cậy trong đánh giá.

Ví dụ:

Điểm Nội dung	Tốt	Đạt	Không đạt
Kiến thức	Trả lời đúng, đủ.	Có một số nhận xét chưa chính xác.	Sai kiến thức.
Kĩ năng	Làm đúng theo hướng dẫn hoặc làm cách khác đúng.	Làm đúng theo hướng dẫn nhưng có một số thao tác còn lỗi.	Không làm theo hướng dẫn.
Thái độ	Hoàn thành mục tiêu chung dựa trên trách nhiệm cá nhân.	Mục tiêu chung chưa hoàn thành hoặc có cá nhân chưa hoàn thành trách nhiệm.	Cả mục tiêu chung và trách nhiệm cá nhân đều không hoàn thành.
Trình bày	Trình bày rõ ràng, mạch lạc.	Trình bày được nội dung nhưng chưa rõ ràng.	Không truyền đạt được nội dung.

PHẦN
HAI

HƯỚNG DẪN DẠY HỌC CÁC BÀI CỤ THỂ

MỘT SỐ ĐIỂM CẦN LƯU Ý VÀ GỢI Ý DẠY HỌC CHUNG

A. GỢI Ý VỀ PHƯƠNG TIỆN

• Đối với các bài học của chuyên đề định hướng CS, GV có thể thực hiện bài giảng tuỳ theo điều kiện vật chất cụ thể của nhà trường. Trừ các tiết học thực hành, HS bắt buộc học và thực hành trên phòng máy, còn tiết lí thuyết thì có nhiều lựa chọn, ví dụ:

– Nếu lớp học không có máy tính thì GV sẽ dạy hoàn toàn trên bảng (viết lệnh mà không cần máy tính).

– Nếu GV có máy tính và máy chiếu thì có thể giảng dạy và thực hiện lệnh trực tiếp trên máy tính, chiếu cho cả lớp theo dõi.

– GV có thể cho phép HS dùng điện thoại thông minh để cài ứng dụng chạy Python trên nền di động. Khi GV thực hiện thao tác trên máy tính, HS có thể làm theo trên điện thoại di động của mình. Trên điện thoại di động có thể cài các phần mềm như QPython 3L hay Pydroid 3.

– Nếu GV dạy lí thuyết trên phòng máy thì HS có thể thực hành ngay các lệnh mà GV thao tác trên máy tính của mình. Cách này tốt nhất vì HS sẽ hiểu ngay bài học tại lớp.

• Trên máy tính GV và máy tính phòng thực hành cần cài bản Python lõi (tải về và cài đặt từ trang: https://www.python.org/downloads/) và tối thiểu một trong các phần mềm soạn thảo lập trình Python. Phần mềm lập trình được khuyên dùng trong các nhà trường là bản Wingware Personal, có thể tải về từ địa chỉ:

https://wingware.com/downloads/wingide-personal

hoặc bản Thonny, tải về từ địa chỉ: https://thonny.org/

• Với bài học liên quan đến thiết kế thuật toán, GV có thể sử dụng hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

B. GỢI Ý VỀ PHƯƠNG PHÁP GIẢNG DẠY

Sau đây là một số gợi ý về cách giảng dạy phần lập trình Python.

1. PHƯƠNG PHÁP DẠY PHẦN LÍ THUYẾT

– GV linh hoạt thực hiện các cách dạy sau:

– Chia nhóm HS, mỗi nhóm 2 – 5 em. Các em có thể đọc sách và thảo luận, hoặc mỗi nhóm có một máy tính bảng đã cài Python, các em thực hành ngay trên máy tính bảng của mình và thực hiện theo yêu cầu của GV.

– GV thao tác một lần trên máy tính (qua máy chiếu) của mình, sau đó gọi một số HS lên bảng thực hiện lại.

– GV gọi một số HS lên bảng thực hiện, cuối cùng GV thao tác lại một lượt để chốt kiến thức.

– GV thao tác trên bảng, trao đổi và hỏi HS các câu hỏi xem HS có hiểu không, sau đó chốt kiến thức.

– Nếu HS có máy tính, GV thao tác trên máy tính của mình và yêu cầu HS thực hiện lại ngay trên máy tính của mình.

– Nếu HS có máy tính bảng hoặc điện thoại thông minh đã cài Python, GV thao tác trên máy tính của mình và yêu cầu HS thao tác lại trên điện thoại của mình.

Phương pháp giảng dạy chung phần kiến thức thuật toán có thể theo quy trình như sau:

1. Giới thiệu bài toán, ý nghĩa thực tế của thuật toán.

2. Mô tả chính xác yêu cầu bài toán, đầu vào, đầu ra của thuật toán.

3. Mô tả, giải thích ý nghĩa các bước thực hiện thuật toán. Tại bước này có thể minh hoạ thuật toán trên các bộ dữ liệu đầu vào đơn giản để học sinh quan sát được.

4. Mô tả chi tiết thuật toán. Có thể mô tả thuật toán theo một trong các cách sau:

(i) Mô tả bằng liệt kê từng bước.

(ii) Mô tả bằng sơ đồ khối.

(iii) Mô tả bằng các lệnh mã giả (pseudocode).

(iv) Mô tả bằng chương trình trên một ngôn ngữ lập trình cụ thể.

5. Thực hiện viết chương trình cài đặt thuật toán đã mô tả ở bước trên.

GV có thể thực hiện và tổ chức giảng dạy tất cả các bước trên theo các cách khác nhau tuỳ thuộc vào tình tình cụ thể tại lớp của mình.

2. PHƯƠNG PHÁP HƯỚNG DẪN THỰC HÀNH

1. GV giới thiệu tổng quan mục đích của bài thực hành, các nhiệm vụ cần thực hiện.

2. Với mỗi nhiệm vụ, GV đọc đề bài, phân tích cách làm, sau đó yêu cầu HS thực hiện dựa trên SGK đã có.

3. Trong quá trình thực hiện nếu HS gặp khó khăn, GV sẽ hướng dẫn và giải quyết các khó khăn đó của HS.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

CHUYÊN ĐỀ 1. THỰC HÀNH THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT ĐỆ QUY

BÀI 1. ĐỆ QUY VÀ HÀM ĐỆ QUY

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Trình bày được tính đệ quy trong một vài định nghĩa sự vật, sự việc.

– Xác định được phần cơ sở và phần đệ quy trong chương trình đệ quy.

– Nhận biết được tính ưu việt của kĩ thuật đệ quy trong định nghĩa và mô tả sự vật cũng như thiết kế chương trình.

– Nhận biết và phân biệt được các khái niệm như công thức truy hồi và chương trình đệ quy.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông; sử dụng được ngôn ngữ lập trình bậc cao, qua đó phát triển tư duy điều khiển và tự động hoá.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

GV: Máy tính đã cài đặt Python và máy chiếu; Hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

HS: Vở và kiến thức Python cơ bản.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động 1	Hoạt động 2	Hoạt động 3
Làm quen với khái niệm đệ quy.	Tìm hiểu công thức truy hồi và các khái niệm liên quan đến đệ quy.	Tìm hiểu và thiết lập hàm đệ quy.

Hoạt động khởi động

Mục đích: HS làm quen với những hiện tượng, sự vật trong cuộc sống có liên quan đến khái niệm đệ quy sẽ học trong bài. GV cho các nhóm HS trao đổi để nhận ra các tính chất chung nhất của các hiện tượng, khái niệm này và tìm thêm nhiều ví dụ khác trong thực tế.

1. KHÁI NIỆM ĐỆ QUY

Hoạt động 1. Làm quen với đệ quy

Mục đích: HS biết và nắm được khái niệm đệ quy.

Tổ chức hoạt động: GV cho HS quan sát hình ảnh giới thiệu của SGK và một số ví dụ về khái niệm đệ quy trong sách. Sau đó yêu cầu các nhóm tìm thêm ví dụ khác từ thực tế để hiểu sâu hơn định nghĩa và ý nghĩa của khái niệm đệ quy. Tìm các ví dụ để thấy nếu không dùng định nghĩa đệ quy thì sẽ rất khó trình bày đúng về khái niệm đó.

Cuối cùng GV chốt kiến thức.

Hoạt động củng cố kiến thức

1. Trường hợp D không có tích chất đệ quy.

2. Đáp án D là sai.

2. CÔNG THỨC TRUY HỒI

Hoạt động 2. Tìm hiểu công thức truy hồi và các khái niệm liên quan đến đệ quy

Mục đích. HS hiểu được khái niệm công thức truy hồi và sự liên quan đến khái niệm đệ quy.

Tổ chức hoạt động: GV chia các nhóm, mỗi nhóm tìm hiểu một dãy số nổi tiếng đã nêu trong SGK. GV có thể đưa thêm nhiều ví dụ khác ngoài sách. Các nhóm tìm hiểu và đưa ra các đặc điểm chung của các dãy số này.

GV chốt kiến thức về công thức truy hồi của các dãy số truy hồi, từ đó nêu ra sự liên hệ với khái niệm đệ quy.

Hoạt động củng cố kiến thức

1. Phần cơ sở: $0! = 1, 1! = 1$.

Phần đệ quy: $n! = n \times (n-1)!$

2. Phần cơ sở: $x^0 = 1$

Phần đệ quy: $x^n = x \times x^{n-1}$

3. HÀM ĐỆ QUY

Hoạt động 3. Tìm hiểu và thiết lập hàm đệ quy

Mục đích. HS hiểu và viết được các hàm đệ quy đơn giản.

Tổ chức hoạt động: GV yêu cầu hai bạn HS lên bảng và thực hiện ngay trên máy tính hai chương trình đã ghi trong phần câu hỏi của hoạt động và chạy thử. Các nhóm cùng trao đổi và tìm ra lỗi của hai chương trình này.

GV cùng sửa hai chương trình để chương trình chạy hết lỗi.

Cuối cùng, GV chốt kiến thức bằng cách mô tả cấu trúc tối thiểu phải có của các hàm đệ quy khi lập trình chạy trên máy tính.

Hoạt động củng cố kiến thức

1. Với chương trình tính số Fibonacci thứ n như sau:

```
1    def Fibonacci(n):
2        if n == 0:
3            return 0
4        elif n == 1:
5            return 1
6        else:
7            return Fibonacci(n - 1) + Fibonacci(n - 2)
```

Các lệnh từ dòng 2 đến 5 là phần cơ sở.

Các lệnh dòng 6, 7 là phần đệ quy.

2. Câu B đúng.

Hoạt động luyện tập

1. Chương trình in và đếm xuôi từ 1 đến 100 trên màn hình.

```
1    def dem(n):
2        if n <= 100:
3            print(n)
4            dem(n+1)
```

Lời gọi hàm chính là **dem**(1).

2. Chương trình tính số Lucas thứ n.

```
1    def Lucas(n):
2        if n == 0:
3            return 2
4        elif n == 1:
5            return 1
6        else:
7            return Lucas(n-1) + Lucas(n-2)
```

Hoạt động vận dụng

1. Chương trình như sau:

```
1 def Pell(n):
2     if n == 0:
3         return 0
4     elif n == 1:
5         return 1
```

```
6       else:
7           return 2*Pell(n-1) + Pell(n-2)
8
9   n = int(input("Nhập số tự nhiên: "))
10  for k in range(n):
11      print(Pell(k), end = " ")
```

2. Chương trình tính số Pell thứ n.

```
1  def Pell(n):
2      if n == 0:
3          return 0
4      elif n == 1:
5          return 1
6      else:
7          return 2*Pell(n-1) + Pell(n-2)
```

BÀI 2. THIẾT KẾ THUẬT TOÁN BẰNG ĐỆ QUY

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Ứng dụng được kĩ thuật đệ quy trong việc thiết kế một vài thuật toán đơn giản.

– Nhận biết được tính ưu việt của kĩ thuật đệ quy trong thiết kế thuật toán.

– Thiết kế được một vài thuật toán đệ quy đơn giản.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông; sử dụng được ngôn ngữ lập trình bậc cao, qua đó phát triển tư duy điều khiển và tự động hoá.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

– *GV:* Máy tính đã cài đặt Python và máy chiếu; Hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

– *HS*: Vở ghi và kiến thức Python cơ bản, kiến thức về đệ quy và hàm đệ quy.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động khởi động

Mục đích: Gợi ý để dẫn đến việc thiết kế chương trình hay hàm đệ quy. Từ ý tưởng của bạn An, GV khơi gợi HS đến ý tưởng của thiết kế đệ quy.

1. Ý TƯỞNG THIẾT KẾ THEO KĨ THUẬT ĐỆ QUY

Hoạt động 1. Tìm hiểu ý tưởng thiết kế thuật toán theo kĩ thuật đệ quy

Mục đích. HS biết được các ý tưởng và thiết kế đệ quy của một số bài toán đơn giản.

Tổ chức hoạt động: GV chia HS thành các nhóm và mỗi nhóm thực hiện đọc, trao đổi và thực hiện giải quyết một bài toán bằng thiết kế thuật toán đệ quy.

Sau khi các nhóm làm xong, GV sẽ hỏi một số câu hỏi, ví dụ vì sao lại thiết kế lập trình đệ quy, các ví dụ này có những điểm gì giống nhau. Cuối cùng, GV chốt kiến thức.

Hoạt động củng cố kiến thức

1. a) Trong hàm sau, phần cơ sở là các lệnh 2, 3, phần đệ quy là lệnh 5.

```
1 def S(n):
2     if n == 0:
3         return 0
4     else:
5         return n + S(n - 1)
```

b) Trong hàm sau, phần cơ sở là các lệnh 2, 3, phần đệ quy là lệnh 5.

```
1 def exp(n):
2     if n == 0:
3         return 0
4     else:
5         return a*exp(a, n - 1)
```

c) Trong hàm sau, phần cơ sở là các lệnh 2, 3, phần đệ quy là lệnh 5.

```
1 def giaithua(n):
2     if n == 0:
3         return 0
4     else:
5         return n*giaithua(n - 1)
```

2. Vì trong tất cả các thuật toán đệ quy đều phải có yêu cầu từ bài toán có kích thước lớn hơn cần đưa về cùng bài toán đó với kích thước nhỏ hơn.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

2. THUẬT TOÁN TÌM KIẾM NHỊ PHÂN

Hoạt động 2. Thiết kế thuật toán đệ quy cho bài toán tìm kiếm nhị phân

Mục đích. HS hiểu được cách thiết kế đệ quy cho bài toán tìm kiếm nhị phân.

Tổ chức hoạt động: Trước tiên, GV trình bày lại thuật toán tìm kiếm nhị phân đã được học trong SGK phần kiến thức thuật toán tìm kiếm, GV đưa các câu hỏi để các nhóm HS trao đổi, thảo luận về đặc điểm của thuật toán này và các gợi ý để có thể dẫn đến lời giải đệ quy của thuật toán. Đặc điểm quan trọng nhất của thuật toán tìm kiếm nhị phân là sau mỗi bước việc tìm kiếm sẽ được thu gọn lại trên miền dữ liệu có độ dài bằng $\frac{1}{2}$ miền tìm kiếm của bước trước. Đây chính là gợi ý quan trọng nhất để dẫn đến ý tưởng thiết kế đệ quy.

GV mô tả thiết kế đệ quy của thuật toán và cùng các nhóm thực hiện cài đặt thuật toán này ngay trên lớp. Cuối cùng GV chốt kiến thức.

Hoạt động củng cố kiến thức

1. Phần cơ sở nằm ở các lệnh 10, 11 khi left > right, khi dãy rỗng.

2. Nếu A = [1, 3, 7, 9], K = 10 thì lời gọi hàm tìm kiếm nhị phân sẽ gọi đệ quy ba lần.

Lần 1, hàm sẽ gọi đệ quy đến dãy [7, 9].

Lần 2, hàm sẽ gọi đệ quy đến dãy [9].

Lần 3, hàm sẽ gọi đệ quy đến dãy [].

Hoạt động luyện tập

1. Chương trình như sau. Chú ý nếu n <= 0 thì đặt SL(n) = 0.

```
1 def SL(n):
2     if n <= 0:
3         return 0
4     else:
5         if n%2 == 1:
6             return n + SL(n-1)
7         else:
8             return SL(n-1)
```

2. Có hai cách viết chương trình này.

Cách 1.

```
1 def in_nguoc(A):
2     if len(A) > 0:
3         print(A[len(A)-1], end = " ")
4     in_nguoc(A[ : len(A)-1])
```

Cách 2. Lời gọi hàm chính: reverse(A,len(A)-1)

```
1 def reverse(A,k):
2     print(A[k], end = " "):
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

```
3       if k > 0:
4           reverse(A,k-1)
```

Lời gọi hàm chính:

```
reverse(A,len(A)-1)
```

Hoạt động vận dụng

1. a) Hàm tính tổng S = 1! + 2! + ... + n! không sử dụng kĩ thuật đệ quy.

```
1 def TGiaiThua(n): # Không sử dụng kĩ thuật đệ quy
2       p = 1
3       S = 0
4       for i in range(1,n+1):
5           p = p*i
6           S = S + p
7       return S
```

b) Hàm tính tổng ở câu a có sử dụng kĩ thuật đệ quy.

```
1 def TGT(p,k,n): # đệ quy
2       if k <= n:
3             p = p*k
4             return p + TGT(p,k+1,n)
5       else:
6             return 0
```

Lời gọi đệ quy của hàm này như sau:

```
TGT(1,1,n)
```

2. Có thể viết lại hàm sắp xếp chèn sử dụng kĩ thuật đệ quy như sau:

```
1 def insertionSortRec(A,i):
2       if i > 0:
3             insertionSortRec(A,i-1)
4             value = A[i]
5             j = i-1
6             while j >= 0 and A[j] > value:
7                   A[j+1] = A[j]
8                   j = j - 1
9                   A[j+1] = value
```

Lời gọi đệ quy của hàm này:

```
insertionSortRec(A,len(A)-1)
```

3. Chương trình của bạn An tìm ra là đúng nếu dãy A không rỗng.

Các lệnh phần cơ sở nằm ở các lệnh từ 2 đến 6.

Chú ý: Chương trình bạn An có thể sửa nhỏ như sau thì chạy sẽ đúng cho tất cả các trường hợp của A (với điều kiện A đã được sắp xếp tăng dần).

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

```
1 def binarySearch(A, left, right, K):
2     if left > right:
3        return -1
4     elif left == right:
5          if A[left] == K:
6              return left
7          else:
8              return -1
9      else:
10         mid = (left + right)//2
11         if A[mid] == K:
12             return mid
13         elif A[mid] < K:
14             return binarySearch(A, mid + 1, right, K)
15         else:
16             return binarySearch(A, left, mid, K)
```

BÀI 3. THỰC HÀNH GIẢI TOÁN THEO KĨ THUẬT ĐỆ QUY

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Viết được một vài chương trình có sử dụng kĩ thuật đệ quy cho một số bài toán đơn giản.

– Viết được chương trình có sử dụng kĩ thuật đệ quy cho một số bài toán đơn giản.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông; sử dụng được ngôn ngữ lập trình bậc cao, qua đó phát triển tư duy điều khiển và tự động hoá.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

– *GV*: Máy tính đã cài đặt Python và máy chiếu; hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

– *HS*: Vở ghi và kiến thức Python cơ bản, kiến thức về đệ quy và hàm đệ quy.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Thực hành
Nhiệm vụ 1. Bài toán biến đổi số nhị phân sang số thập phân. Nhiệm vụ 2. Tính ước chung lớn nhất của hai số nguyên không âm.

Hoạt động khởi động

Mục đích: Nhắc lại cho HS nhớ một số đặc điểm chính của kĩ thuật lập trình đệ quy từ các bài học trước.

Hoạt động thực hành: Nhiệm vụ 1

GV nêu nhiệm vụ, hướng dẫn chung về định hướng thuật toán, HS sẽ thực hiện thiết kế chi tiết và lập trình cho bài toán. Chú ý công thức quan trọng nhất là công thức (2) tính giá trị số thập phân cần tính thông qua biểu diễn nhị phân ban đầu. Từ công thức (2) dẫn đến công thức truy hồi (3).

Hoạt động thực hành: Nhiệm vụ 2

GV nêu nhiệm vụ tính ước chung lớn nhất của hai số nguyên không âm (không đồng thời bằng 0). Công thức quan trọng nhất để tính hàm gcd(a, b) là:

1. Nếu b = 0 thì gcd(a, 0) = a.

2. Nếu b > 0 thì gcd(a, b) = gcd(b, a mod b).

Từ các công thức trên dễ dàng thiết kế được thuật toán đệ quy cho bài toán này.

Hoạt động luyện tập

1. Các bước tính gcd(93,60) như sau.

Bước 1. gcd(93,60) = gcd(60, 33)

Bước 2. gcd(60,33) = gcd(33,27)

Bước 3. gcd(33,27) = gcd(27,6)

Bước 4. gcd(27,6) = gcd(6,3)

Bước 5. gcd(6,3) = gcd(3,0) = 3

2. Chương trình này tương tự chương trình của Nhiệm vụ 1.

```
1 def decimal(A,n):
2     if n == 0:
3         return 0
4     elif n == 1:
```

```
5           return A[0]
6       else:
7           return 2*decimal(A,n-1) + A[n-1]
```

Hoạt động vận dụng

1. Chương trình sử dụng kĩ thuật đệ quy như sau:

```
1 def gcd(a,b):
2       if a == b:
3           return a
4       else:
5           if a > b:
6               return gcd(a-b,b)
7           else:
8               return gcd(a,b-a)
```

2. Chương trình không sử dụng kĩ thuật đệ quy tính gcd() theo thuật toán Euclid.

```
1 def gcd(a,b):
2       while b != 0:
3           r = a % b
4           a = b
5           b = r
6       return a
```

3. Chương trình như sau:

```
1       fi = "chieucao.inp"
2       fo = "ketqua.out"
3
4       def NhapDL(fi):
5           DS = []
6           CC = []
7           f = open(fi,encoding="UTF-8")
8           for line in f:
9               A = line.split()
10              ten = " ".join(A[ : len(A)-1])
11              h = float(A[len(A)-1])
12              DS.append(ten)
13              CC.append(h)
14           f.close()
15           return DS,CC
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

```
16
17      def minmax(A,k):
18          if k == 0:
19              return A[0],A[0]
20          else:
21              x,y = minmax(A,k-1)
22              return min(x,A[k]), max(y,A[k])
23
24      def tinhtoan(DS,CC):
25          Min, Max = minmax(CC,len(CC)-1)
26          imin = CC.index(Min)
27          imax = CC.index(Max)
28          return imin, imax
29
30      def Ghikq(fo,DS,CC,imin,imax):
31          f = open(fo,"w",encoding="UTF-8")
32          print(DS[imin],CC[imin],file = f)
33          print(DS[imax],CC[imax],file = f)
34          f.close()
35
36      # Chương trình chính
37      DS,CC = NhapDL(fi)
38      imin,imax = tinhtoan(DS,CC)
39      Ghikq(fo,DS,CC,imin,imax)
```

BÀI 4. BÀI TOÁN THÁP HÀ NỘI

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Biết và trình bày được lời giải bài toán Tháp Hà Nội sử dụng kĩ thuật đệ quy.

– Thực hiện được trình bày và lập trình giải bài toán Tháp Hà Nội.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

Với bài học này cần các công cụ trực quan để giúp HS dễ hiểu hơn khi tiếp cận bài toán này. Có thể dùng các công cụ mô tả các đĩa và cọc, hoặc dùng hình ảnh, hoặc dùng phần mềm mô phỏng. Dưới đây là một link mô phỏng như vậy:

https://scratch.mit.edu/projects/781207755/

GV: Máy tính đã cài đặt Python và máy chiếu; Hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

HS: Vở ghi và kiến thức Python cơ bản, kiến thức về đệ quy và hàm đệ quy.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động 1	Hoạt động 2	Hoạt động 3
Tìm hiểu bài toán Tháp Hà Nội	Tìm hiểu ý tưởng thiết kế đệ quy cho lời giải bài toán Tháp Hà Nội	Thiết lập thuật toán và cài đặt chương trình cho bài toán Tháp Hà Nội

Hoạt động khởi động

Mục đích: Giới thiệu lịch sử bài toán Tháp Hà Nội xuất hiện vào đầu thế kỉ XX tại Việt Nam như thế nào, tạo hứng thú cho HS tiếp cận bài toán này.

1. MÔ TẢ BÀI TOÁN THÁP HÀ NỘI

Hoạt động 1. Tìm hiểu bài toán Tháp Hà Nội

Mục đích. HS biết được lời giải tường minh bằng hình ảnh của bài toán Tháp Hà Nội với các trường hợp nhỏ, n = 1, 2, 3.

Tổ chức hoạt động: GV cho HS xem sơ đồ lời giải bài toán với các trường hợp nhỏ n = 1, 2, 3 hoặc cho HS xem chương trình mô phỏng lời giải bài toán này, từ đó HS có thể trao đổi, thảo luận để đưa ra các nhận xét ban đầu của mình.

Hoạt động củng cố kiến thức

1. Mô tả bằng lời lời giải bài toán Tháp Hà Nội: chuyển n đĩa từ cọc 1 sang cọc 3, lấy cọc 2 làm trung gian.

- Với n = 1, chỉ cần 1 bước.
- Với n = 2, cần 2 bước.

Chuyển đĩa 1 từ cọc 1 sang cọc 2

Chuyển đĩa 2 từ cọc 1 sang cọc 3

Chuyển đĩa 1 từ cọc 2 sang cọc 3

• Với n = 3, cần 6 bước.

Chuyển đĩa 1 từ cọc 1 sang cọc 3

Chuyển đĩa 2 từ cọc 1 sang cọc 2

Chuyển đĩa 1 từ cọc 3 sang cọc 2

Chuyển đĩa 3 từ cọc 1 sang cọc 3

Chuyển đĩa 1 từ cọc 2 sang cọc 1

Chuyển đĩa 2 từ cọc 2 sang cọc 3

Chuyển đĩa 1 từ cọc 1 sang cọc 3

2. Bài toán chuyển n đĩa từ cọc 1 sang cọc 2, lấy cọc 3 làm trung gian.

• Với n = 2

Chuyển đĩa 1 từ cọc 1 sang cọc 3

Chuyển đĩa 2 từ cọc 1 sang cọc 2

Chuyển đĩa 1 từ cọc 3 sang cọc 2

• Với n = 3.

Chuyển đĩa 1 từ cọc 1 sang cọc 2

Chuyển đĩa 2 từ cọc 1 sang cọc 3

Chuyển đĩa 1 từ cọc 2 sang cọc 3

Chuyển đĩa 3 từ cọc 1 sang cọc 2

Chuyển đĩa 1 từ cọc 3 sang cọc 1

Chuyển đĩa 2 từ cọc 3 sang cọc 2

Chuyển đĩa 1 từ cọc 1 sang cọc 2

2. Ý TƯỞNG LỜI GIẢI BÀI TOÁN THÁP HÀ NỘI

Hoạt động 2. Tìm hiểu ý tưởng thiết kế đệ quy cho lời giải bài toán Tháp Hà Nội

Mục đích. HS hiểu được ý tưởng thiết kế đệ quy của lời giải bài toán Tháp Hà Nội.

Tổ chức hoạt động: GV lần lượt cho các nhóm lên trình bày lời giải chi tiết cho các trường hợp n = 1, 2, 3, từ đó rút ra quy luật đệ quy. GV chốt kiến thức đưa ra sơ đồ lời giải tổng quát.

Hoạt động củng cố kiến thức

Sơ đồ lời giải cho n = 4. Chuyển n đĩa từ cọc 1 sang cọc 3, lấy cọc 2 làm trung gian.

1 : 1 --> 3

2 : 1 --> 2

1 : 3 --> 2

3 : 1 --> 3

1 : 2 --> 1

2 : 2 --> 3

1 : 1 --> 3

4 : 1 --> 2

1 : 3 --> 2

2 : 3 --> 1

1 : 2 --> 1

3 : 3 --> 2

1 : 1 --> 3

2 : 1 --> 2

1 : 3 --> 2

3. THIẾT LẬP CHƯƠNG TRÌNH GIẢI BÀI TOÁN THÁP HÀ NỘI

Hoạt động 3. Thiết lập thuật toán và cài đặt chương trình cho bài toán Tháp Hà Nội

Mục đích. HS biết cách thiết kế thuật toán và chương trình đệ quy giải bài toán Tháp Hà Nội.

Tổ chức hoạt động: GV cho các nhóm đọc tài liệu và tự lập trình cho lời giải bài toán. Cuối cùng, GV chốt kiến thức và chương trình ở cả dạng mã giả và lập trình Python.

Hoạt động củng cố kiến thức

1. H(2) = 3, H(3) = 7, H(4) = 15, H(5) = 31

2. Chương trình tính H(n) có thể như sau, sử dụng công thức đệ quy:

$$H(1) = 1, H(n) = 2H(n-1) + 1$$

```
1 def H(n):
2     if n == 1:
3         return 1
4     else:
5         return 2*H(n-1) + 1
```

Hoạt động luyện tập

1. Vẫn sử dụng chương trình cũ nhưng với lời gọi khác như sau:

```
Hanoi(n,"A","B","C")
```

2. Chương trình cho kết quả đúng.

Hoạt động vận dụng

1. Chứng minh công thức $H(n) = 2^n - 1$ bằng quy nạp.

• Với n = 1, ta đã biết $H(1) = 1 = 2^1 - 1$. Vậy công thức trên đúng với n = 1.

• Giả sử công thức trên đúng với n, ta sẽ chứng minh nó đúng với n+1. Ta có:

$H(n+1) = 2H(n) + 1$

• Theo giả thiết quy nạp ta có: $H(n) = 2^n - 1$, từ đó suy ra:

$H(n+1) = 2H(n) + 1 = 2(2^n - 1) + 1 = 2^{n+1} - 1$.

Vậy công thức đúng với n + 1.

2. Lời giải có thể như sau:

```
def Hanoi(n,i,j,k):
    if n == 1:
        return [(n,i,j)]
    else:
        return Hanoi(n-1,i,k,j) + [(n,i,j)] + Hanoi(n-1,k,j,i)
```

BÀI 5. THỰC HÀNH THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT ĐỆ QUY

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

Thực hành thiết kế một số bài toán sử dụng kĩ thuật đệ quy.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

– *GV:* Máy tính đã cài đặt Python và máy chiếu; Hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

– *HS*: Vở ghi và kiến thức Python cơ bản, kiến thức về đệ quy và hàm đệ quy.

C PHƯƠNG PHÁP GIẢNG DẠY CHUNG

– GV giới thiệu tổng quan mục đích của bài thực hành, các nhiệm vụ cần thực hiện.

– Với mỗi nhiệm vụ, GV đọc đề bài, phân tích cách làm, sau đó yêu cầu HS thực hiện dựa trên SGK đã có.

– Trong quá trình thực hiện nếu HS gặp khó khăn, GV sẽ hướng dẫn và giải quyết các khó khăn đó của HS.

D GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Thực hành
Nhiệm vụ 1. Thuật toán sắp xếp chèn. Nhiệm vụ 2. Chuyển số thập phân sang số nhị phân.

Hoạt động khởi động

Mục đích: Ôn lại một số đặc điểm, ưu điểm và nhược điểm của kĩ thuật đệ quy trong lập trình.

Chú ý: Ưu điểm: ngắn gọn, sáng sủa, dễ hiểu; Nhược điểm: tốc độ chạy chậm.

Hoạt động thực hành: Nhiệm vụ 1

GV mô tả lại ý tưởng gốc của thuật toán sắp xếp chèn từ đó gợi ý sự liên quan đến ý tưởng đệ quy và dẫn đến lời giải đệ quy.

Hoạt động thực hành: Nhiệm vụ 2

Với bài này, GV giới thiệu mô hình chuyển sang số nhị phân bằng cách tính tay theo sơ đồ của bảng 5.1, từ đó dẫn đến lời giải đệ quy.

GV chú ý với HS hai điều sau:

a) Nếu bỏ qua các lệnh 4, 5 thì kết quả sẽ luôn chèn chữ số 0 vào đầu xâu kết quả, dẫn đến không chính xác, vì vậy cần có thêm các lệnh 4, 5.

b) Nếu bỏ qua lệnh 4, 5 và lệnh 3 thay thế bằng lệnh **return “”** thì kết quả sẽ đúng với mọi n > 0 nhưng sai với n = 0.

Hoạt động luyện tập

1. Chương trình này tương tự như lời giải của Nhiệm vụ 2.

```
1 def binary (n):
2     if n == 0:
```

```
3           return [0]
4       elif n == 1:
5           return [1]
6       else:
7           return binary(n//2) + [n%2]
```

2. Chương trình có thể như sau:

```
1 def decimal(s,n):
2       if n == 0:
3           return 0
4       elif n == 1:
5           return int(s[0])
6       else:
7           return 2*decimal(s,n-1) + int(s[n-1])
```

Lời gọi đệ quy là:

```
decimal(s,len(s))
```



Hoạt động vận dụng

1. Chương trình đếm số nghịch đảo của dãy số A cho trước.

a) Phương án không sử dụng kĩ thuật đệ quy.

```
1 def nghich_dao(A):
2       C = 0
3       n = len(A)
4       for i in range(n-1):
5           for j in range(i,n):
6               if A[i] > A[j]:
7                   C = C + 1
8           return C
```

b) Phương án sử dụng kĩ thuật đệ quy

```
1 def nghich_dao(A,n):
2       if n <= 0:
3           return 0
4       else:
5           C = 0
6           for i in range(n):
7               if A[i] > A[n]:
8                           C = C + 1
9           return C + nghich_dao(A,n-1)
```

Lời gọi đệ quy của hàm là:

```
nghich_dao(A,len(A)-1)
```

2. Giả sử A = [a_0, a_1, ..., a_n] là dãy các hệ số của đa thức F(A,x,n).

$F(A, x, n) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$

Chương trình đệ quy tính F(A,x,n) như sau:

```
1 def F(A,x,k):
2     if k == 0:
3         return A[0]
4     else:
5         return A[k]*(x**k) + F(A,x,k-1)
```

Lời gọi đệ quy của hàm là:

```
F(A,x,len(A)-1)
```

CHUYÊN ĐỀ 2. THỰC HÀNH THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT CHIA ĐỂ TRỊ

BÀI 6. Ý TƯỞNG VÀ KĨ THUẬT CHIA ĐỂ TRỊ

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Biết và giải thích được ý tưởng của kĩ thuật chia để trị thông qua một số bài toán đơn giản.

– Nêu được mối quan hệ giữa kĩ thuật chia để trị và đệ quy.

– Thực hiện được tìm kiếm nhị phân bằng kĩ thuật đệ quy dưới ý tưởng của chia để trị.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

– *GV:* Có thể tìm hình ảnh hoặc công cụ hỗ trợ mô phỏng trò chơi tìm viên bi giả.

– *HS*: Vở ghi và kiến thức Python cơ bản, kiến thức về đệ quy và hàm đệ quy.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động 1	Hoạt động 2	Hoạt động 3
Tìm hiểu ý tưởng của kĩ thuật chia để trị để giải một bài toán.	Tìm hiểu kĩ thuật chia để trị thông qua tìm kiếm nhị phân.	Đánh giá độ phức tạp thời gian của thuật toán tìm kiếm nhị phân.

Hoạt động khởi động

Mục đích: Giới thiệu bài toán – trò chơi tìm viên bi giả. Cách giải bài toán này sẽ dẫn đến ý tưởng của phương pháp chia để trị là nội dung kiến thức chính của bài học.

1. Ý TƯỞNG CHIA ĐỂ TRỊ

Hoạt động 1. Tìm hiểu ý tưởng của kĩ thuật chia để trị để giải một bài toán

Mục đích. Tìm hiểu lời giải bài toán – trò chơi tìm viên bi giả, từ đó dẫn đến mô hình ý tưởng của kĩ thuật chia để trị.

Tổ chức hoạt động:

– GV cho các nhóm tìm hiểu lời giải của bài toán tìm viên bi giả, các nhóm có thể trình bày hoặc biểu diễn cách chơi.

– Sau đó, GV chốt lại cách chơi và từ cách giải bài toán này dẫn đến phương pháp chia để trị.

– Cuối cùng, GV chốt kiến thức.

Hoạt động củng cố kiến thức

1. Với 9 viên bi, số lần cân tối đa là 3 lần.

2. Mô tả chi tiết cách cân của 9 viên bi.

Bước 1. Phân tách số bi thành $4 + 4 + 1$ và cân 2 nhóm 4 viên. Nếu cân cân bằng thì tìm ngay 1 viên bi kia là bi giả. Nếu cân không cân bằng thì lấy ra 4 viên bên nặng hơn và thực hiện tiếp Bước 2.

Bước 2. Thực hiện tiếp đệ quy giải bài toán với 4 viên bi đã chọn. Kết quả của bước này sẽ tìm được ngay viên bi giả.

2. THUẬT TOÁN TÌM KIẾM NHỊ PHÂN

Hoạt động 2. Tìm hiểu kĩ thuật chia để trị thông qua tìm kiếm nhị phân

Mục đích. HS biết và hiểu được cách hiểu mới của thuật toán tìm kiếm nhị phân theo kĩ thuật chia để trị.

Tổ chức hoạt động:

– GV nhắc lại phương án đệ quy của bài toán tìm kiếm nhị phân đã được học trong Bài 2 của chuyên đề này và diễn giải thuật toán này bằng phương pháp chia để trị.

– Cuối cùng, GV chốt kiến thức.

Hoạt động củng cố kiến thức

1. Phần cơ sở là các lệnh tại dòng 10, 11.

2. Trường hợp left = right các bước thực hiện như sau:

Bước 1. Tính mid = left.

Bước 2. Nếu A[mid] = K thì trả về mid.

Bước 3. Nếu A[mid] ≠ K thì;

Bước 3.1. Nếu mid < K thì gọi đệ quy đến A[mid+1 .. right] và trả về -1.

Bước 3.2. Nếu mid > K thì gọi đệ quy đến A[left .. mid – 1] và trả về -1.

Hoạt động 3. Đánh giá độ phức tạp thời gian của thuật toán tìm kiếm nhị phân

Mục đích. HS biết được độ phức tạp thời gian của thuật toán tìm kiếm nhị phân và nhận biết được tính ưu việt của phương pháp chia để trị.

Tổ chức hoạt động:

– Về đánh giá độ phức tạp thời gian của thuật toán tìm kiếm nhị phân T(n) = O(logn), GV có thể trình bày nhanh như đã mô tả trong sách hoặc có thể công nhận không chứng minh kết quả này.

– Tuy nhiên, GV cần nhấn mạnh kết quả T(n) = O(logn) của tìm kiếm nhị phân so với tìm kiếm tuần tự là T(n) = O(n), là một bậc thời gian tốt hơn hẳn. Qua đó thấy được tính ưu việt hơn hẳn của tìm kiếm nhị phân, tức là chia để trị, so với tìm kiếm tuần tự, tức là tư duy tự nhiên.

Hoạt động củng cố kiến thức

1. Nếu dãy gốc có 1 phần tử thì số phép toán đơn tối đa là 4 (xem lời giải câu hỏi củng cố 2 của Hoạt động 2).

2. Trường hợp dãy gốc có 2 phần tử. Số phép toán đơn được tính như sau:

– Các dòng 2, 3, 4 mỗi dòng cần thực hiện 1 phép tính.

– Các dòng tiếp theo sẽ gọi đệ quy tại dòng 7, 9 sẽ sử dụng nhiều nhất là 4 phép tính nữa (theo bài 1 ở trên).

Vậy, tổng số phép tính đơn nhiều nhất là 3 + 4 = 7.

Hoạt động luyện tập

1. Chương trình có thể như sau (không kiểm tra dãy đầu vào):

```
def binarySearch(A, left, right, K):
    if left <= right:
        mid = (left + right)//2
        if A[mid] == K:
            return mid
        elif A[mid] < K:
            return binarySearch(A, mid + 1, right, K)
        else:
            return binarySearch(A, left, mid - 1, K)
    else:
        return -1

# Chương trình chính
s = input("Nhập dãy số nguyên tăng dần: ")
A = [int(x) for x in s.split()]
K = int(input("Nhập số cần tìm: "))
if binarySearch(A,0,len(A)-1,K) >= 0:
        print("Tìm thấy")
else:
        print("Không tìm thấy")
```

2. Chương trình như sau:

```
def binarySearch(A, left, right, K):
    if left <= right:
        mid = (left + right)//2
        if A[mid] == K:
            return mid
        elif A[mid] < K:
            return binarySearch(A, mid + 1, right, K)
        else:
            return binarySearch(A, left, mid - 1, K)
    else:
        return -1

    # Chương trình chính
    DS = [["Bình", 7.5], ["Hoa", 8], ["An", 9], ["Quang", 10]]
    Ten = [x[0] for x in DS]
    Diem = [x[1] for x in DS]
```

```
17      K = float(input("Nhập một điểm số: "))
18      index = binarySearch(Diem,0,len(Diem)-1,K)
19      if index >= 0:
20         print("Tìm thấy bạn", Ten[index])
21      else:
22         print("Không tìm thấy")
```

Hoạt động vận dụng

1. Phương án không đệ quy của tìm kiếm nhị phân vẫn chính là chia để trị, nhưng sử dụng kĩ thuật không đệ quy của chia để trị.

2. Chương trình như sau:

```
1 from random import randint
2 from time import perf_counter
3 def sinhDay(n):
4     A = []
5     for i in range(n):
6         A.append(randint(1,n))
7     return A
8
9 def binarySearch(A, left, right, K):
10    if left <= right:
11       mid = (left + right)//2
12       if A[mid] == K:
13            return mid
14       elif A[mid] < K:
15            return binarySearch(A, mid + 1, right, K)
15       else:
16            return binarySearch(A, left, mid - 1, K)
18    else:
19       return -1
20
21 def LinearSearch(A,K):
22     for i in range(len(A)):
23         if A[i] == K:
24              return i
25     return -1
26
27      # Chương trình chính
28      n = 1000
29      A = sinhDay(n)
```

```
30      B = sorted(A)
31      K = randint(1,n)
32      t1 = perf_counter()
33      LinearSearch(A,K)
34      t2 = perf_counter()
35      print("Tìm kiếm tuần tự:",t2-t1)
36      t1 = perf_counter()
37      binarySearch(B,0,len(B)-1,K)
38      t2 = perf_counter()
39      print("Tìm kiếm nhị phân:",t2-t1)
```

3. Hàm chính của thuật toán theo chia để trị như sau:

```
1 def floorSqrt(n,start,end):
2     if start == end:
3         return start
4         mid = (start+end)//2
5         if mid*mid == n:
6             return mid
7         elif mid*mid < n:
8             if (mid+1)*(mid+1) > n:
9                 return mid
10            return floorSqrt(n,mid+1,end)
11    else:
12        return floorSqrt(n,start,mid-1)
```

Lời gọi đệ quy là:

```
floorSqrt(n,0,n)
```

BÀI 7. THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT CHIA ĐỂ TRỊ

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Biết và thực hiện được một số thiết kế thuật toán bằng kĩ thuật chia để trị.

– Thực hành được thiết kế một số bài toán sử dụng kĩ thuật chia để trị đơn giản.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

B CHUẨN BỊ

– *GV:* Máy tính đã cài đặt Python và máy chiếu; Hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

– *HS*: Vở ghi và kiến thức Python cơ bản, kiến thức về kĩ thuật chia để trị.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động 1	Hoạt động 2
Giải bài toán tính luỹ thừa bằng chia để trị.	Giải bài toán tìm kiếm nhị phân mở rộng bằng chia để trị.

Hoạt động khởi động

Mục đích: Giới thiệu các bài toán sẽ được học thiết kế trong bài.

1. BÀI TOÁN TÍNH LUỸ THỪA

Hoạt động 1. Giải bài toán tính luỹ thừa bằng chia để trị

Mục đích. HS hiểu được cách thiết lập thuật toán tính luỹ thừa nhanh bằng chia để trị.

Tổ chức hoạt động:

– Trước tiên, GV mô tả bài toán và yêu cầu các nhóm giải bài này bằng cách tính thông thường, bằng tay và chương trình.

– Tiếp theo, GV sẽ giới thiệu ý tưởng mới giải bài toán tính luỹ thừa theo kĩ thuật chia để trị.

– Sau khi thực hiện xong việc cài đặt chương trình, GV cho các nhóm thảo luận nhanh về độ phức tạp thời gian của thuật toán.

– GV chốt kiến thức về độ phức tạp thời gian của thuật toán.

Hoạt động củng cố kiến thức

1. Tính tay 2^{11}.

– Tính theo phương pháp cũ. Cần thực hiện 10 lần phép nhân với 2.

– Tính theo phương pháp chia để trị mới.

Bước 1. Tính $2 \times 2 = 2^2$

Bước 2. Tính $4 \times 4 = 2^4$

Bước 3. Tính $2^4 \times 2 = 2^5$

Bước 4. Tính $2^5 \times 2^5 = 2^{10}$

Bước 5. Tính $2^{10} \times 2 = 2^{11}$

Vậy, theo cách tính mới chỉ cần 5 phép nhân.

2. Tính a^{21}.

$a^{21} = a \times a^{10} \times a^{10}$

Theo bài trên, tính a^{10} cần 4 phép nhân, vậy suy ra cần 4 + 2 = 6 phép nhân cho a^{21}.

2. BÀI TOÁN TÌM KIẾM NHỊ PHÂN MỞ RỘNG

Hoạt động 2. Giải bài toán tìm kiếm nhị phân mở rộng bằng chia để trị

Mục đích. HS hiểu được bài toán và cách giải theo kĩ thuật chia để trị của bài toán tìm kiến nhị phân mở rộng.

Tổ chức hoạt động:

– GV mô tả bài toán và yêu cầu các nhóm trao đổi, thực hiện bài toán theo cách tìm kiếm tuần tự, tức là thuật toán tự nhiên.

– Sau đó, GV sẽ chuyển sang cách làm mới, thiết kế theo phương pháp chia để trị, cách làm này sẽ là một mở rộng của thuật toán tìm kiếm nhị phân đã biết từ các bài trước.

– Cuối cùng, GV chốt kiến thức.

Hoạt động củng cố kiến thức

1. Dòng 4 mô tả trường hợp A có một phần tử thì đáp số luôn là phần tử duy nhất này. Dòng 6 mô tả trường hợp A có hai phần tử, đáp án sẽ là phần tử nào gần K hơn về giá trị.

2. Chương trình trên khác thuật toán tìm kiếm nhị phân như sau:

– Cách xử lí các trường hợp cơ sở khi A có 1 hoặc 2 phần tử.

– Sau mỗi bước lặp (hoặc sau mỗi lần gọi đệ quy), phạm vi tìm kiếm sẽ tìm tiếp ở một trong hai khoảng [**mid, right**] hay [**left, mid**], điều này khác biệt với tìm kiếm nhị phân là sau mỗi vòng lặp miền tìm kiếm sẽ là [**left, mid-1**] hoặc [**mid+1, right**].

Hoạt động luyện tập

1. Chương trình không đệ quy cho bài toán tìm kiếm nhị phân mở rộng như sau:

```
1 def closevalue(A,k):
2     left = 0
3     right = len(A) - 1
4     while left <= right:
5         if left == right:
6             return A[left]
7         elif left == right - 1:
8             if abs(A[left] - K) < abs(A[right] - K):
9                 return A[left]
10            else:
```

```
            return A[right]
        else:
            mid = (left + right)//2
            if A[mid] == K:
                return A[mid]
            elif K < A[mid]:
                right = mid
            else:
                left = mid
```

2. Bài luyện tập này tương tự nhiều bài khác.



Hoạt động vận dụng

1. Thuật toán tính a^n theo chia để trị nhưng không sử dụng kĩ thuật đệ quy như sau:

```
def exp(a,n):
    p = 1
    c = a
    while n != 0:
        if n % 2 == 0:
            n = n//2
            c = c*c
        else:
            n = n - 1
            p = p*c
    return p
```

2. Bài toán tính vùng chỉ số như sau:

```
def findRange(A,left,right,K):
    if left > right:
        return -1,-1
    else:
        mid = (left + right)//2
        if A[mid] == K:
            start = mid
            while start >= left and A[start] == K:
                start = start - 1
            end = mid
            while end <= right and A[end] == K:
                end = end + 1
            return start+1, end-1
        elif A[mid] < K:
            return findRange(A,mid+1,right,K)
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

```
16          else:
17              return findRange(A,left,mid-1,K)
```

Lời gọi đệ quy chính của hàm như sau:

```
findRange(A,0,len(A)-1,K)
```

BÀI 8. THỰC HÀNH THIẾT KẾ THUẬT TOÁN TÌM KIẾM THEO KĨ THUẬT CHIA ĐỂ TRỊ

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Biết cách thiết kế và viết chương trình giải một số bài toán theo kĩ thuật chia để trị.

– Thực hành viết chương trình giải một số bài toán theo kĩ thuật chia để trị.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển được năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

– *GV*: Với những bài mô tả thiết kế thuật toán, GV có thể sử dụng hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản. Hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

– *HS*: Vở ghi và kiến thức Python cơ bản, kiến thức về kĩ thuật chia để trị.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động khởi động

Mục đích: Giới thiệu các bài toán sẽ được học thiết kế trong bài.

Hoạt động thực hành

GV giới thiệu bài toán và yêu cầu HS đọc, trình bày và thực hiện cài đặt phương án giải đầu tiên của bài toán với thuật toán tìm kiếm tuần tự. GV nhấn mạnh cho HS biết cách giải này sẽ có độ phức tạp thời gian O(n), với n là kích thước dãy đầu vào.

Tiếp theo là phần chính của bài thực hành. GV giới thiệu ý tưởng chia để trị của bài toán dựa trên thuật toán tìm kiếm nhị phân.

GV phân công các nhóm tìm hiểu, sau đó trình bày thuật toán.

GV giải thích nhanh cách tính thời gian của thuật toán này và kết luận: Độ phức tạp thời gian của thuật toán chỉ là O(logn), nhanh hơn hẳn so với cách tìm bằng phương pháp tìm kiếm tuần tự.

Cuối cùng cả lớp tiến hành lập trình giải bài toán.

Hoạt động luyện tập

Chỉ cần chỉnh sửa chương trình gốc tại hai vị trí sau.

```
1 def countNum(A,left,right,C):
2     if left > right:
3         return None, None, 0
4     else:
5         mid = (left+right)//2
6         if A[mid] == C:
7             start = mid
8             while start >= left and A[start] == C:
9                 start = start - 1
10            end = mid
11            while end <= right and A[end] == C:
12                end = end + 1
13            return start+1, end-1, end - start - 1
14        elif A[mid] < C:
15            return countNum(A,mid+1,right,C)
16        else:
17            return countNum(A,left,mid-1,C)
```

Các vị trí cần thay đổi so với chương trình gốc.

Hoạt động vận dụng

1. Bài tập này có thể giải bằng phương pháp chia để trị như sau, chú ý rằng phương pháp này thực hiện trên dãy số bất kì (không là đơn điệu tăng, giảm) nên tổng thời gian thực hiện chương trình của lời giải này vẫn là O(n), tương đương với cách làm bằng tìm kiếm tuần tự.

Hàm đệ quy được xây dựng như sau:

```
1 def find(A,left,right,K):
2     if left > right:
3         return 0
4     else:
5         mid = (left + right)//2
6         if A[mid] == K:
7             count = 1
8         else:
9             count = 0
10        return count + find(A,left,mid-1,K) + find(A,mid+1,right,K)
```

2. Đây là bài toán không khó nhưng cần chú ý khi phân tích và tìm thuật toán giải. Để bài nằm trong bài học phần chia để trị nên cần tìm cách giải theo hướng này. Xét các trường hợp quan trọng sau:

a) Trường hợp dãy A tăng thực sự, tức là giả sử dãy A được sắp xếp thoả mãn điều kiện sau:

A[0] < A[1] < .... < A[n-1].

Trường hợp này có thể thiết kế thuật toán hoàn toàn tương tự với thuật toán tìm kiếm nhị phân đã biết. Điều này được suy ra từ phân tích sau:

Hàm đệ quy cần thiết lập là fixvalue(A,left,right). Hàm sẽ trả lại chỉ số của phần tử cố định nếu tồn tại, ngược lại sẽ trả về -1.

– Đặt mid = (left + right)//2. Nếu A[mid] = mid thì hiển nhiên mid chính là phần tử cố định và chương trình sẽ dừng lại và trả về giá trị mid.

– Nếu A[mid] < mid. Khi đó có thể chứng minh rằng bên trái của mid không thể tồn tại bất kì phần tử cố định nào nữa.

Thật vậy, vì A là đơn điệu tăng thực sự nên A[mid-1] < A[mid].

Từ đó suy ra: A[mid - 1] ≤ A[mid] - 1 < mid - 1.

Vậy mid - 1 không thể là phần tử cố định. Tương tự, suy ra mọi phần tử bên trái của mid không là cố định.

Từ điều trên suy ra trong trường hợp này, phần tử cố định, nếu có, chỉ có thể nằm ở bên phải giá trị mid.

– Nếu ngược lại A[mid] > mid thì suy luận tương tự suy ra bên phải của mid không thể có phần tử cố định, do đó phần tử cố định chỉ có thể có ở bên trái của mid.

Các suy luận trên sẽ dẫn đến lời giải sau của bài toán. Trong trường hợp này, sau mỗi bước, hàm sẽ gọi đệ quy đến mảng có kích thước bằng $\frac{1}{2}$ mảng gốc, do đó có thể tính công thức thời gian chạy của chương trình như sau:

T(n) = T(n/2) + O(1)

Từ đó suy ra T(n) = O(logn).

Lời giải của bài toán trong trường hợp này như sau:

```
def fixvalue(A,left,right):
    if left > right:
        return -1
    else:
        mid = (left+right)//2
        if A[mid] == mid:
            return mid
        elif A[mid] < mid:
            return fixvalue(A,mid+1,right)
```

```
10          else:
11              return fixvalue(A,left,mid-1)
```

b) Trường hợp A là dãy được sắp xếp tăng dần, nhưng không thực sự.

$A[0] \le A[1] \le .... \le A[n-1]$.

Trường hợp này, không thể giải theo cách đã nêu trên. Do vậy, có thể thiết lập lời giải cải tiến có thời gian chạy O(n) như sau:

```
1 def findFixPoint(A,left,right):
2     if left > right:
3         return -1
4     else:
5         mid = (left+right)//2
6         if A[mid] == mid:
7             return mid
8         if mid+1 <= A[right]:
9             index = findFix(A,mid+1,right)
10             if index >= 0:
11                 return index
12         if mid - 1 >= A[left]:
13             index = findFix(A,left,mid-1)
14             if index >= 0:
5                  return index
16         return -1
```

c) Trường hợp A bất kì thì có thể thiết lập một lời giải hoàn toàn theo kĩ thuật chia để trị nhưng thời gian chạy là O(n). Lời giải sau thậm chí có thể áp dụng cho mảng A bất kì, không cần là đơn điệu tăng.

```
1 def findFix(A,left,right):
2     if left > right:
3         return -1
4     else:
5         mid = (left+right)//2
6         if A[mid] == mid:
7              return mid
8         index = findFix(A,mid+1,right)
9         if index >= 0:
10            return index
11        index = findFix(A,left,mid-1)
12        if index >= 0:
13            return index
14        return -1
```

3. Các hàm firstInd() và lastInd() có thể được thiết lập dễ dàng như sau. Cả hai hàm này đều được thiết kế tương tự như thuật toán tìm kiếm nhị phân và đều có thời gian chạy O(logn).

Hàm firstInd(A,left,right,C) trả về chỉ số phần tử đầu tiên của A[left,right] có giá trị bằng C, ngược lại trả về -1.

```
1 def firstInd(A,left,right,C):
2     if left <= right:
3         mid = (left+right)//2
4         if A[mid] == C:
5             i = mid
6             while i >= left and A[i] == C:
7                 i = i - 1
8             return i+1
9         elif A[mid] < C:
10            return firstInd(A,mid+1,right,C)
11        else:
12            return firstInd(A,left,mid-1,C)
13    else:
14        return -1
```

Khi đó hàm *countNum*(A,left,right,C) có thể tính như sau:

`countNum(A,left,right,C) = lastInd(A,left,right,C) - firstInd(A,left,right,C) + 1.`

BÀI 9. SẮP XẾP TRỘN

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Biết và trình bày được cách thiết kế thuật toán sắp xếp trộn bằng kĩ thuật chia để trị.

– Thực hiện được lập trình theo thuật toán sắp xếp trộn.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

B CHUẨN BỊ

– *GV*: Máy tính đã cài đặt Python và máy chiếu; Hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

– *HS*: Vở ghi và kiến thức Python cơ bản, kiến thức về kĩ thuật chia để trị.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động 1	Hoạt động 2	Hoạt động 3
Tìm hiểu ý tưởng sắp xếp trộn.	Tìm hiểu các bước thực hiện sắp xếp trộn.	Tìm hiểu đánh giá độ phức tạp thời gian của sắp xếp trộn.

Hoạt động khởi động

Mục đích: Hướng sự tò mò của học sinh muốn tìm hiểu một thuật toán sắp xếp khác tốt hơn hẳn các thuật toán sắp xếp đã biết như thuật toán sắp xếp chèn, sắp xếp chọn, sắp xếp nổi bọt. Kĩ thuật hướng đến chính là chia để trị.

1. Ý TƯỞNG THUẬT TOÁN SẮP XẾP TRỘN

Hoạt động 1. Tìm hiểu ý tưởng sắp xếp trộn

Mục đích. HS hiểu được ý tưởng chính, gốc của sắp xếp trộn.

Tổ chức hoạt động:

– GV cho các nhóm HS quan sát và thực hiện ý tưởng chính của sắp xếp trộn, tư tưởng lấy từ sơ đồ 9.1. Tạm thời chưa cần tìm hiểu quá chi tiết về thuật toán trộn, nhưng GV có thể cho HS biết nhanh ý nghĩa của việc trộn: Trộn hai dãy đã sắp xếp thành một dãy lớn hơn cũng sẽ được sắp xếp.

– GV giao cho các nhóm nghiên cứu và mô tả ý tưởng của sắp xếp trộn theo ba bước *Chia, Trị và Kết hợp*.

– GV chốt kiến thức chính là ý tưởng chia để trị của sắp xếp trộn.

Hoạt động củng cố kiến thức

1. Thực hiện bằng tay trộn hai dãy B = 1, 4, 7 và C = 2, 3, 6.

Mỗi bước sau mô tả một thao tác đơn thực hiện trộn hai dãy B, C, kết quả trả về dãy mới A.

Bước 1. Đưa 1 vào A.

Bước 2. Đưa 2 vào A

Bước 3. Đưa 3 vào A

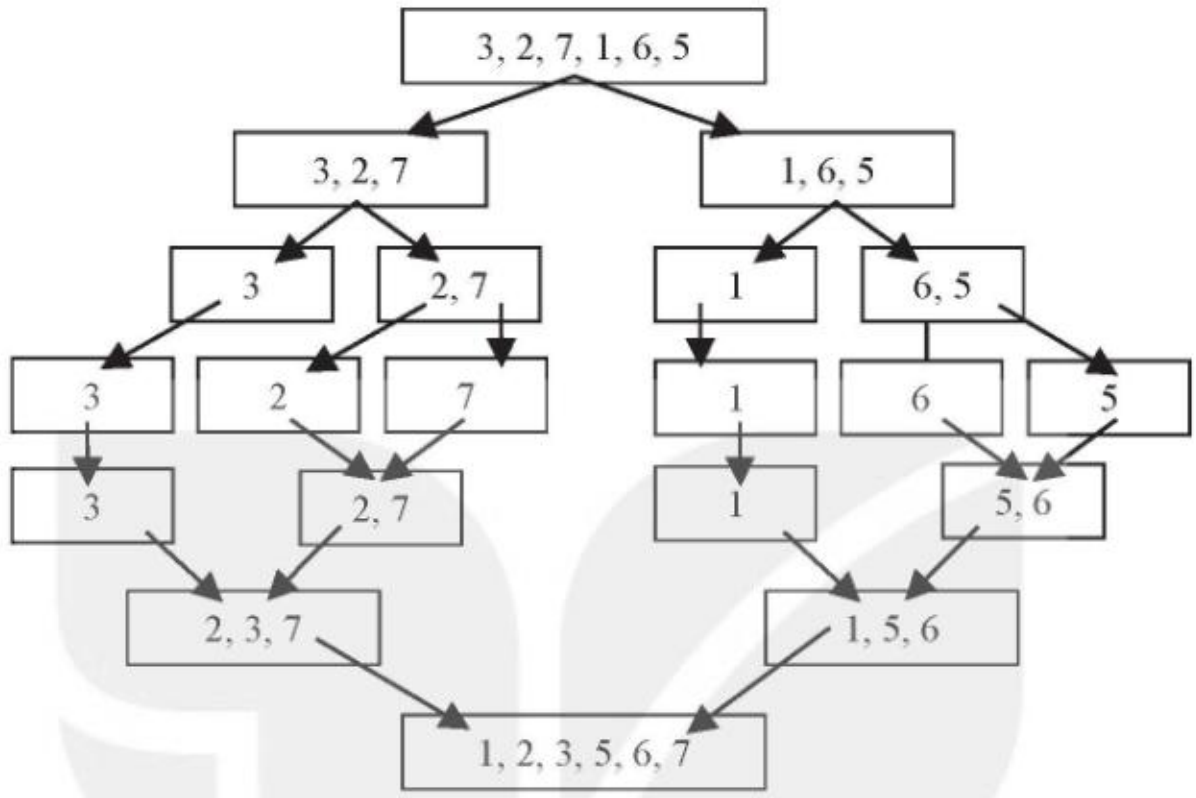
Bước 4. Đưa 4 vào A

Bước 5. Đưa 6 vào A

Bước 6. Đưa 7 vào A.

Kết thúc dãy A sẽ là 1, 2, 3, 4, 6, 7.

2. Thực hiện sắp xếp trộn dãy 3, 2, 7, 1, 6, 5 theo sơ đồ 9.1.



Sơ đồ 9.1

2. MÔ TẢ THUẬT TOÁN SẮP XẾP TRỘN

Hoạt động 2. Tìm hiểu các bước thực hiện sắp xếp trộn

Mục đích. HS hiểu rõ các bước chính của sắp xếp trộn.

Tổ chức hoạt động. GV mô tả (hoặc cho các nhóm tự tìm hiểu, sau đó GV chốt lại) các bước chính của sắp xếp trộn. Có thể hiểu sắp xếp trộn luôn bao gồm hai quá trình:

(1) Tách dãy gốc thành hai dãy con với mỗi dãy con có kích thước $\frac{1}{2}$ dãy ban đầu. Dùng đệ quy để sắp xếp hai dãy con này.

(2) Sau đó, dùng thuật toán trộn để trộn hai dãy con đã sắp xếp thành dãy ban đầu nhưng đã được sắp xếp.

Thuật toán sắp xếp trộn được mô tả tổng quát như sau:

```
1 def mergeSort(A,left,right):
2     if left < right:
3         mid = (left + right)//2
4         mergeSort(A,left,mid)
5         mergeSort(A,mid+1,right)
6         merge(A,left,mid,right)
```

Trong đó lệnh 6 chính là thuật toán "trộn" đóng vai trò rất quan trọng của sắp xếp trộn.

GV cần phân tích kĩ ý tưởng này để HS hiểu được bản chất và ý nghĩa của sắp xếp trộn.

Hoạt động củng cố kiến thức

1. Vòng lặp 6 có nhiều nhất là m + n bước lặp.

2. Nếu dãy gốc có 1 phần tử thì thuật toán sắp xếp trộn sẽ không thực hiện gì và trả về ngay dãy gốc.

3. ĐÁNH GIÁ THUẬT TOÁN SẮP XẾP TRỘN

Hoạt động 3. Tìm hiểu đánh giá độ phức tạp thời gian của thuật toán sắp xếp trộn

Mục đích. HS biết được tính ưu việt của sắp xếp trộn.

Tổ chức hoạt động: GV phân tích và làm rõ cho HS hiểu được ý nghĩa quan trọng của sắp xếp trộn là có thời gian chạy chỉ là O(nlogn), nhanh hơn hẳn các thuật toán sắp xếp đã biết, đều có thời gian chạy $O(n^2)$.

Hoạt động củng cố kiến thức

1. Tính thời gian chạy thuật toán sắp xếp trộn với A = [3, 1]. Chúng ta sẽ phân tích và tính theo số đơn vị thời gian.

– Kiểm tra tại dòng 2, mất 1 đơn vị thời gian.

– Thực hiện lệnh tại dòng 5, mất 1 đơn vị thời gian.

– Các dòng 5, 6 mỗi dòng chỉ mất 1 đơn vị thời gian do B và C đều là mảng 1 phần tử.

– Dòng 8 sẽ mất 2 đơn vị thời gian.

Do vậy tổng cộng mất 6 đơn vị thời gian.

2. Các bước thực hiện sắp xếp trộn với A = [5, 1, 7, 4] như sau:

Bước 1 - Dòng 5, tính k = 2.

Bước 2 - Tính B = [5,1]

Bước 3 - Tính C = [4,7]

Bước 4 - Trộn B, C và trả về dãy [1,4,5,7]

Thời gian chạy tính như sau:

– Kiểm tra tại dòng lệnh 1, mất 1 đơn vị thời gian.

– (1) tốn 1 đơn vị thời gian.

– (2), (3) mỗi bước mất 6 đơn vị thời gian (xem câu 1).

– (4) tốn 4 đơn vị thời gian.

Tổng cộng sẽ mất 2 + 12 + 4 = 18 đơn vị thời gian.

Hoạt động luyện tập

1. Chương trình như sau:

```
1 fi = "Data.inp"
2 def NhapDL(fi):
3       f = open(fi)
4       B = [int(x) for x in f.readline().split()]
5       C = [int(x) for x in f.readline().split()]
6       f.close()
7       return B,C
8
9 def merge(B,C):
10    m = len(B)
11    n = len(C)
12    A = [0]*(m+n)
13    i = j = k = 0
14    while i < m and j < n:
15        if B[i] < C[j]:
16            A[k] = B[i]
17            i = i + 1
18        else:
19            A[k] = C[j]
20            j = j + 1
21        k = k + 1
22    if i == m:
23        while j < n:
24            A[k] = C[j]
25            j = j+1
26            k = k+1
27    else:
28        while i < m:
29            A[k] = B[i]
30            i = i + 1
31            k = k + 1
32     return A
33
34 # Chương trình chính
35 B,C = NhapDL(fi)
36 A = merge(B,C)
37 print(A)
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

2. Chương trình có thể như sau:

```
fi = "Data1.inp"
def NhapDL(fi):
    f = open(fi)
    A = [int(x) for x in f.readline().split()]
    f.close()
    return A

def merge(B,C):
    m = len(B)
    n = len(C)
    A = [0]*(m+n)
    i = j = k = 0
    while i < m and j < n:
        if B[i] < C[j]:
            A[k] = B[i]
            i = i + 1
        else:
            A[k] = C[j]
            = j + 1
        k = k + 1
    if i == m:
        while j < n:
            A[k] = C[j]
            j = j+1
            k = k+1
     else:
        while i < m:
            A[k] = B[i]
            i = i + 1
            k = k + 1
     return A

def mergeSort(A):
    if len(A) <= 1:
        return A
    else:
        k = len(A)//2
        B = mergeSort(A[0:k])
        C = mergeSort(A[k:len(A)])
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

```
40          return merge(B,C)
41
42 # Chương trình chính
43 A = NhapDL(fi)
44 print(A)
45 A = mergeSort(A)
46 print(A)
```

Hoạt động vận dụng

1. Chương trình sau sẽ so sánh sắp xếp chèn với sắp xếp trộn.

```
1 from time import perf_counter
2 fi = "Data2.inp"
3 def NhapDL(fi):
4     f = open(fi)
5     A = [int(x) for x in f.readline().split()]
6     f.close()
7     return A
8
9 def merge(B,C):
10    m = len(B)
11    n = len(C)
12    A = [0]*(m+n)
13    i = j = k = 0
14    while i < m and j < n:
15        if B[i] < C[j]:
16            A[k] = B[i]
17            i = i + 1
18        else:
19            A[k] = C[j]
20            j = j + 1
21        k = k + 1
22    if i == m:
23        while j < n:
24            A[k] = C[j]
25            j = j+1
26            k = k+1
27     else:
28        while i < m:
29            A[k] = B[i]
30            i = i + 1
31            k = k + 1
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

```
    return A

def mergeSort(A):
    if len(A) <= 1:
        return A
    else:
            k = len(A)//2
            B = mergeSort(A[0:k])
            C = mergeSort(A[k:len(A)])
        return merge(B,C)

def InsertionSort(A):
     n = len(A)
     for i in range(1,n):
         value = A[i]
         j = i-1
         while j >= 0 and A[j] > value:
             A[j+1] = A[j]
             j = j - 1
         A[j+1] = value

# Chương trình chính
A = NhapDL(fi)
B = A.copy()
t1 = perf_counter()
mergeSort(A)
t2 = perf_counter()
print("Sắp xếp trộn:",t2 - t1)

t1 = perf_counter()
InsertionSort(B)
t2 = perf_counter()
print("Sắp xếp chèn:",t2-t1)
```

2. Sắp xếp trộn tại chỗ. Sau đây là một phương án. Chú ý rằng có rất nhiều phương án khác nhau thực hiện việc sắp xếp trộn tại chỗ.

```
def merge(A,left,mid,right):
    B = [0]*len(A)
    i = left
    j = mid + 1
    k = left
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

```
    while i <= mid and j <= right:
        if A[i] < A[j]:
            B[k] = A[i]
            i = i + 1
        else:
            B[k] = A[j]
            j = j + 1
        k = k + 1
    if i == mid+1:
        while j <= right:
            B[k] = A[j]
            j = j + 1
            k = k + 1
    else:
        while i <= mid:
            B[k] = A[i]
            i = i + 1
            k = k + 1
    A[left:right+1] = B[left:right+1]

def mergeSort(A,left,right):
    if left < right:
    mid = (left + right)//2
    mergeSort(A,left,mid)
    mergeSort(A,mid+1,right)
    merge(A,left,mid,right)
```

BÀI 10. THỰC HÀNH GIẢI TOÁN BẰNG KĨ THUẬT CHIA ĐỂ TRỊ

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Biết cách thiết kế và viết chương trình giải một số bài toán theo kĩ thuật chia để trị.

– Thực hành viết chương trình giải một số bài toán theo kĩ thuật chia để trị.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển được năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

– *GV*: Với những bài mô tả thiết kế thuật toán, GV có thể sử dụng hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản. Hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

– *HS*: Vở ghi và kiến thức Python cơ bản, kiến thức về kĩ thuật chia để trị.

C PHƯƠNG PHÁP GIẢNG DẠY CHUNG

Phương pháp giảng dạy chung phần kiến thức thuật toán có thể theo quy trình như sau:

1. Giới thiệu bài toán, ý nghĩa thực tế của thuật toán.

2. Mô tả chính xác yêu cầu bài toán, đầu vào, đầu ra của thuật toán.

3. Mô tả, giải thích ý nghĩa các bước thực hiện thuật toán. Tại bước này có thể minh hoạ thuật toán trên các bộ dữ liệu đầu vào đơn giản để học sinh quan sát được.

4. Mô tả chi tiết thuật toán.

5. Thực hiện viết chương trình cài đặt thuật toán đã mô tả ở bước trên.

Tất cả các bước trên GV có thể thực hiện và tổ chức giảng dạy theo các cách khác nhau tuỳ thuộc vào tình tình cụ thể tại lớp của mình.

Phương pháp chung giảng dạy phần thực hành:

1. GV giới thiệu tổng quan mục đích của bài thực hành, các nhiệm vụ cần thực hiện.

2. Với mỗi nhiệm vụ, GV đọc đề bài, phân tích cách làm, sau đó yêu cầu HS thực hiện dựa trên SGK đã có.

3. Trong quá trình thực hiện nếu HS gặp khó khăn, GV sẽ hướng dẫn và giải quyết các khó khăn đó của HS.

D GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động khởi động

Mục đích: Đặt vấn đề nhận biết một dãy đã sắp xếp tăng dần hay giảm dần hay chưa sẽ dẫn đến bài toán tìm cặp nghịch đảo. GV cùng HS có thể dẫn đến: Dãy số đã được sắp xếp tăng dần nếu không có cặp nghịch đảo nào, ngược lại dãy số sẽ sắp xếp giảm dần nếu nó có $\frac{n(n-1)}{2}$ cặp nghịch đảo. Do vậy bài toán đếm cặp nghịch đảo cũng là một cách dùng để nhận biết một dãy số đã sắp xếp tăng dần hoặc giảm dần hay chưa.

Hoạt động thực hành

– GV giới thiệu bài toán chính, nhắc lại cho HS nhớ, đây chính là bài toán quen thuộc HS đã biết từ Bài 26 của SGK Tin học 11 – Định hướng Khoa học máy tính. Bài tập này đã được làm mẫu để học thiết kế theo phương pháp làm mịn dần.

– GV nhắc lại lời giải bài toán đã được học trong SGK, HS cùng thực hiện lời giải này.

– GV phân tích lời giải đã biết thì thời gian chạy của chương trình là $O(n^2)$.

– Sau đó, GV giới thiệu cách thiết kế mới của bài toán theo kĩ thuật chia để trị. Giới thiệu ý tưởng lời giải của bài toán là theo cách làm của thuật toán sắp xếp trộn đã học ở Bài 9.

– GV cho HS trao đổi để hiểu ý tưởng và cách làm của bài toán, sau đó HS tiến hành cài đặt chương trình.

– GV chốt lại ưu điểm của thuật toán này chỉ chạy với thời gian O(nlogn), nhanh hơn hẳn so với cách làm đã biết.

Hoạt động luyện tập

Với bài tập này chúng ta chỉ cần sửa một chút ở hai hàm chính là *merge*(A,left,mid,right) và *getInvCount*(A,left,right). Kết quả mới của hai hàm này là một dãy các cặp phần tử nghịch đảo.

Hàm *merge*(A,left,mid,right) sẽ trả về dãy các cặp nghịch đảo khi tiến hành "trộn" tại chỗ hai phần của cùng dãy A. Phần thay đổi chương trình nằm ở dòng lệnh 13. Chú ý kết quả trả về dãy kq các cặp phần tử nghịch đảo sau khi thực hiện lệnh *merge*().

```
1  def merge(A,left,mid,right):
2      i = left
3      j = mid + 1
4      k = left
5      kq = []
6      while i <= mid and j <= right:
7          if A[i] <= A[j]:
8              temp_arr[k] = A[i]
9              k = k + 1
10             i = i + 1
11         else:
12             temp_arr[k] = A[j]
13             kq.extend([(x,A[j]) for x in A[i:mid+1]])
14             k = k + 1
15             j = j + 1
16     while i <= mid:
17         temp_arr[k] = A[i]
18         k = k + 1
19         i = i + 1
20     while j <= right:
21         temp_arr[k] = A[j]
22         k = k + 1
23         j = j + 1
24     for l in range(left, right + 1):
```

```
25          A[l] = temp_arr[l]
26      return kq
```

Hàm chính *getInvCount*(A,left,right) sẽ trả về kết quả là dãy các cặp phần tử nghịch đảo của dãy A trong đoạn chỉ số [left, right].

```
1 def getInvCount(A,left,right):
2     if left >= right:
3         return []
4     else:
5         mid = (left + right)//2
6         L = getInvCount(A,left,mid)
7         R = getInvCount(A,mid + 1,right)
8         M = merge(A,left,mid,right)
9         return L + M + R
```

Đoạn chương trình chính có thể như sau:

```
1 # Chương trình chính
2 A = [4, 5, 2, 10, 4]
3 n = len(A)
4 temp_arr = [0]*n
5 inv_array = getInvCount(A,0,n-1)
6 print(inv_array)
```

Chú ý: Với yêu cầu mới này thì hàm merge() sẽ có thời gian chạy là $O(n^2)$, do đó công thức tính thời gian của bài toán sẽ là:

$T(n) = 2T(n/2) + O(n^2)$.

Từ công thức trên có thể tính được $T(n) = O(n^2)$.

Vậy bài toán tìm tất cả các cặp phần tử nghịch đảo làm theo kĩ thuật chia để trị không có thời gian tốt hơn là cách làm thông thường.

Hoạt động vận dụng

1. Lời giải bài toán tính phần tử mode trong trường hợp dãy A được sắp xếp theo thứ tự tăng dần như sau:

```
1 def mode(A,left,right):
2     " Tìm 1 phần tử mode, A đã được sắp xếp tăng dần "
3     if left > right:
4         return None,0
5     else:
6         mid = (left+right)//2
7         i = mid
8         while i > left and A[i] == A[i-1]:
9             i = i - 1
```

```
        j = mid
        while j < right and A[j] == A[j+1]:
            j = j + 1
        m0,k0 = A[mid],j-i+1
        m1,k1 = mode(A,left,mid-1)
        m2,k2 = mode(A,mid+1,right)
        if k0 >= k1 and k0 >= k2:
            return m0,k0
        if k1 >= k0 and k1 >= k2:
            return m1,k1
        if k2 >= k0 and k2 >= k1:
            return m2,k2
```

Chú ý: Bài toán tìm phần tử mode trong trường hợp dãy A cho trước bất kì (không là đơn điệu tăng) thì vẫn có thể áp dụng phương pháp chia để trị để giải bài toán này, tuy nhiên sẽ khó hơn. GV có thể đưa bài toán này cho HS như một đề tài nghiên cứu để HS thực hiện.

2. Bài toán tính tổng con có giá trị lớn nhất của dãy số A cho trước.

Hàm đệ quy áp dụng chia để trị có dạng *maxSubarray*(A,left,right) như sau:

```
MinINF = float("-inf") # Số bé vô cùng
def maxSubarray(A,left,right):
    if left > right:
        return None, None, MinINF
    elif left == right:
        return left, right, A[left]
    else:
        mid = (left+right)//2
        maxL = mid
        Sum = 0
        for i in range(mid,left-1,-1):
            Sum = Sum + A[i]
            if Sum > sumL:
                sumL = Sum
                maxL = i
        sumR = A[mid+1]
        maxR = mid+1
        Sum = 0
        for j in range(mid+1,right+1):
            Sum = Sum + A[j]
            if Sum > sumR:
                sumR = Sum
                maxR = j
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

```
        sumX = sumL + sumR
        leftX, rightX = maxL,maxR
        leftL, rightL, sumL = maxSubarray(A,left,mid)
        leftR, rightR, sumR = maxSubarray(A,mid+1,right)
        if sumL >= sumR and sumL >= sumX:
            return leftL, rightL, sumL
        elif sumR >= sumL and sumR >= sumX:
            return leftR, rightR, sumR
        else:
            return leftX, rightX, sumX
```

Gọi T(n) là thời gian chạy của hàm đệ quy chính của chương trình. Thời gian tính các tham số leftX, rightX, sumX sẽ là O(n). Do đó, chúng ta có công thức:

$T(n) = 2T(n/2) + O(n)$.

Từ công thức trên sẽ suy ra: $T(n) = O(nlogn)$.

Chương trình chính có thể như sau:

```
# Chương trình chính
A = [13, -3, -25, 20, -3, -16, -23, 18, 20, -7, 12, -5, -22, 15, -4, 7]
left,right,Sum = maxSubarray(A,0,len(A)-1)
if Sum > MinINF:
    print("Dãy con cực đại có tổng là:",Sum)
    for k in range(left,right+1):
        print(A[k],end = " ")
else:
    print("Dãy rỗng")
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

CHUYÊN ĐỀ 3. THỰC HÀNH THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT DUYỆT

BÀI 11. BÀI TOÁN TÌM KIẾM VÀ KĨ THUẬT DUYỆT

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Nêu được ý tưởng của kĩ thuật duyệt, tìm kiếm vét cạn và ví dụ minh hoạ.

– Hiểu và thực hiện được kĩ thuật duyệt, tìm kiếm vét cạn.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển được năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

GV: Máy tính đã cài đặt Python và máy chiếu; Hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

HS: Vở và kiến thức Python cơ bản.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động khởi động

Mục đích: HS biết hiểu và nắm được ý tưởng của kĩ thuật duyệt và tìm kiếm vét cạn. GV có thể lấy nhiều ví dụ hơn về các bài toán duyệt trong thực tế, ví dụ tìm kiếm học sinh trong danh sách học sinh trong lớp, tìm kiếm sử dụng nhiều từ khoá/điều kiện với Google search,....

1. Kĩ thuật duyệt trong bài toán tìm kiếm

Hoạt động 1. Kĩ thuật duyệt trong bài toán tìm kiếm

Mục đích: HS hiểu được kĩ thuật duyệt trong bài toán tìm kiếm.

Tổ chức hoạt động: GV chia HS thành các nhóm, mỗi nhóm sẽ thảo luận và tìm cách thực hiện một nhiệm vụ tìm kiếm học sinh trên tập dữ liệu cho trước:

– Tìm một HS có điểm Toán lớn hơn điểm Vật lí.

– Tìm tất cả các HS có điểm Vật lí lớn hơn điểm Hoá học.

– Tìm tất cả các HS có cả ba điểm đều lớn hơn hoặc bằng 9.

GV giúp các em HS hiểu được sự khác nhau giữa yêu cầu tìm một HS thoả mãn yêu cầu và tìm tất cả các HS thoả mãn yêu cầu cho trước.

Cuối cùng, GV chốt kiến thức theo SGK.

Hoạt động củng cố kiến thức

1. So sánh số vòng lặp cần thực hiện để thực hiện các yêu cầu a, b, c trong ví dụ trên:

Trong yêu cầu a, nếu duyệt từ đầu dãy đến cuối dãy thì chương trình có thể kết thúc sau vòng lặp thứ hai, ngay khi phát hiện HS đầu tiên có điểm Toán lớn hơn điểm Vật lí. Với yêu cầu b và yêu cầu c, chương trình yêu cầu tìm tất cả các HS thoả mãn yêu cầu cho trước nên chúng ta cần duyệt toàn bộ HS có trong danh sách, số vòng lặp cần thực hiện là 6.

2. Phân tích và viết chương trình để thực hiện các yêu cầu sau:

a) Tìm HS có điểm Toán bằng 8.9.

b) Tìm một HS có tổng điểm ba môn Toán, Vật lí, Hoá học lớn hơn 26.5.

c) Tìm HS có tổng điểm ba môn Toán, Vật lí, Hoá học nhỏ nhất.

Với yêu cầu a) chúng ta có thể duyệt từ đầu dãy và dừng lại khi tìm thấy một HS có điểm Toán bằng 8.9

```
for i in range(len(A)):
    if B[i]==8.9:
            kqa=A[i]
            break
if kqa != “”
    print(“Tên một học sinh có điểm toán bằng 8.9 là:”, kqa)
```

Yêu cầu b được thực hiện tương tự yêu cầu a, ta có thể duyệt từ đầu dãy và dừng lại khi tìm thấy HS có tổng điểm ba môn lớn hơn 26.5.

```
for i in range(len(A)):
    if B[i]+ C[i]+ D[i] > 26.5:
            kqb=A[i]
            break
if kqb != “”
      print(“Tên một học sinh có tổng điểm 3 môn lớn hơn 26.5 là:”, kqb)
```

Yêu cầu c để bài yêu cầu tìm HS có tổng điểm ba môn nhỏ nhất, ta cần phải duyệt toàn bộ danh sách các HS trong danh sách.

```
min_score = B[0] + C[0] + D[0]
kqc = A[0]
for i in range(len(A)):
    if B[i]+ C[i]+ D[i] < min_score:
            min_score= B[i]+ C[i]+ D[i]
            kqc = A[i]
print(“Tên một học sinh có tổng điểm 3 môn lớn hơn 26.5 là:”, kqc)
```

2. Tìm kiếm vét cạn

Hoạt động 2. Tìm kiếm vét cạn

Mục đích: HS hiểu và thực hiện được ý tưởng của kĩ thuật tìm kiếm vét cạn.

Tổ chức hoạt động: GV chia HS thành các nhóm, mỗi nhóm sẽ thảo luận về cách thức thực hiện bài toán 1 hoặc bài toán 2

Trước khi HS thảo luận, GV có thể giúp các em học sinh hiểu được xâu nhị phân là gì, cũng như nhắc lại kiến thức toán học về hoán vị.

Cuối cùng, GV chốt kiến thức theo SGK.

Hoạt động củng cố kiến thức

1. Trong kĩ thuật tìm kiếm tuần tự, chúng ta phải duyệt toàn bộ các phần tử có trong mảng dữ liệu, do đó tìm kiếm tuần tự được coi là một ví dụ của duyệt vét cạn.

2. Mảng hai chiều kích thước m × n có tổng số m × n phần tử. Theo định nghĩa của duyệt vét cạn, chúng ta cần duyệt toàn bộ các phần tử có trong mảng, số phần tử phải duyệt là m × n.

3. Thuật toán tìm kiếm nhị phân chỉ duyệt một vài phần tử trong mảng dữ liệu nên không phải kĩ thuật duyệt vét cạn.

Hoạt động luyện tập

1. Để thực hiện bài toán này, chúng ta cần duyệt tất cả các số nguyên dương nhỏ hơn N để tìm ra số có nhiều ước số nhất.

```
def uoc_so(N):
    result = 0
    for i in range(1,N+1):
        if N%i ==0:
            result = result  +1
    return result
N = int(input("Nhập số nguyên dương N: "))
uoc_so_max = 1
kq = 1
for i in range(1,N):
    if uoc_so (i)> uoc_so_max:
        uoc_so_max = uoc_so (i)
        kq = i
print("Số nhỏ hơn N có số ước số lớn nhất là :",kq)
```

2. Bài toán này có thể thực hiện bằng cách sau: sử dụng các thuật toán sắp xếp đã học để sắp xếp danh sách HS theo thứ tự tăng hoặc giảm dần, 3 HS cần tìm là 3 HS ở cuối hoặc đầu dãy.

```
def SelectionSort(A,B):
    n = len(A)
    for i in range(n-1):
        iMax = i
        for j in range(i+1,n):
            if A[j] > A[iMax]:
                iMax = j
        A[i],A[iMax] = A[iMax],A[i]
        B[i],B[iMax] = B[iMax],B[i]

```

```
A = [‘Nam’, ‘Sơn’, ‘Hương’,’Huyền’, ‘Hà’, ‘Hùng’]
B = [ 8.3, 8.4, 8.7, 8.9, 9.1, 9.6]
C = [ 8.3, 7.8, 8.9, 9.5, 9.3, 9.0]
D = [7.9, 9.0, 8.9, 8.2, 9.5, 9.1]
E=[0]*len(A)
for i in range(0,len(A)):
    E[i] = B[i] + C[i] + D[i]
SelectionSort(E,A) # Sắp xếp theo thứ tự tổng điểm giảm dần
print(“3 học sinh có tổng điểm cao nhất là:”, A[0],A[1],A[2])
```

Hoạt động vận dụng

1. Với bài toán này, ta cần duyệt tất cả các phần tử từ đầu dãy đến phần tử thứ hai tính từ cuối dãy để liệt kê tất cả các bộ ba phần tử thoả mãn điều kiện là ba số nguyên liên tiếp.

```
A = [10,11,12, 4, 2, 6,7,8, 7,3,1,10,9,8,7,6]
kq =[]
number = 0
for i in range(0,len(A)-2):
    if A[i] == A[i+1] + 1 and A[i+1] == A[i+2] + 1:
        kq.append((A[i],A[i+1],A[i+2]))
        number = number + 1
    if A[i] == A[i+1] - 1 and A[i+1] == A[i+2] - 1:
        kq.append((A[i],A[i+1],A[i+2]))
        number = number + 1
print(“Số các bộ số thoả mãn điều kiện là:”,number)
print(“Các bộ số thoản mãn điều kiện là:”,kq)
```

2. Để giải quyết bài toán này, ta cần viết chương trình kiểm tra một số có phải số hoàn hảo hay không. Sau đó, duyệt tất cả các số từ 1 đến N-1 để xem đó có phải số hoàn hảo hay không.

```
def perfect_number(N):
    sum = 0
    for i in range(1,N):
        if N%i ==0:
            sum = sum  +i
    if sum == N:
        return True
    else:
        return False
N = int(input(“Nhập số nguyên dương N: ”))
kq =[]
for i in range(1,N):
    if perfect_number(i)==True:
```

```
14          kq.append(i)
15 if kq!="":
16     print("Các số hoàn hảo nhỏ hơn N là:",kq)
```

BÀI 12. THỰC HÀNH KĨ THUẬT DUYỆT CHO BÀI TOÁN TÌM KIẾM

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Áp dụng được kĩ thuật duyệt để giải quyết một vài bài toán tìm kiếm cụ thể.

– Hiểu và thực hiện được kĩ thuật duyệt để giải quyết bài toán ví dụ về tìm bạn chung và đề xuất kết bạn.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển được năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

GV: Máy tính đã cài đặt Python và máy chiếu; Hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

HS: Vở và kiến thức Python cơ bản.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động khởi động

Mục đích: Nhắc lại về ý nghĩa và ý tưởng của kĩ thuật duyệt đã học trong bài trước. GV có thể liên hệ với ứng dụng Facebook, xem kĩ thuật duyệt có thể được thực hiện trong các tác vụ nào.

Hoạt động thực hành: Nhiệm vụ 1

Mục đích: HS biết được cách sử dụng kĩ thuật duyệt để tìm ra danh sách bạn bè của đối tượng cần tìm.

Tổ chức hoạt động: GV chia HS thành các nhóm, mỗi nhóm cùng thảo luận xem bài toán tính số bạn chung cần thực hiện thế nào, sau đó có thể so sánh cách làm của nhóm mình và cách làm được hướng dẫn trong sách khác nhau như thế nào? Cách làm nào tối ưu hơn?

GV có thể giải thích qua về hai bước sẽ phải thực hiện để giải quyết bài toán này trước khi cho HS bắt đầu thảo luận nhóm.

Sau khi kết thúc thảo luận, cho các nhóm tiến hành viết, biên dịch, và chạy thử chương trình.

Hoạt động thực hành: Nhiệm vụ 2

Mục đích: HS biết được cách sử dụng kĩ thuật vét cạn để tìm ra toàn bộ danh sách đề xuất kết bạn.

Tổ chức hoạt động: GV chia HS thành các nhóm, mỗi nhóm cùng thảo luận xem bài toán tìm danh sách đề xuất kết bạn có thể thực hiện thế nào, sau đó có thể so sánh cách làm của nhóm mình và cách làm được hướng dẫn trong sách khác nhau như thế nào? Cách làm nào tối ưu hơn?

GV có thể giải thích qua về hai bước sẽ phải thực hiện để giải quyết bài toán này trước khi cho HS bắt đầu thảo luận nhóm.

Hoạt động luyện tập

Để in ra danh sách bạn chung của hai tài khoản, trong hàm tính số bạn chung của hai tài khoản, ngoài số bạn chung, chúng ta cần trả về thêm một mảng chứa danh sách bạn chung của hai tài khoản đó. Chương trình trong Nhiệm vụ 1 có thể được sửa lại như sau:

```
1 def banchung (A,B): # tính số phần tử chung có trong hai danh sách
2     kq = 0
3     banchung=[]
4     for i in range(len(A)):
5         if A[i] in B:
6             kq= kq+1
7             banchung.append(A[i])
8     return kq,banchung
```

Sau khi tìm được hai tài khoản Facebook cần kiểm tra, chúng ta có thể sử dụng hàm tính số bạn chung như sau:

```
9 kq,ban_list = banchung(ban_fb[index1],ban_fb[index2])
10 print("Số bạn chung của hai tài khoản facebook là: ",kq,ban_list)
```

Hoạt động vận dụng

1. Để giải quyết được bài toán này, chúng ta phải lần lượt thực hiện các bước sau:

– Thống kê được danh sách các nhóm.

– Viết chương trình tính số thành viên hiện có trong nhóm.

– Sau khi nhập tên người dùng cần gợi ý, duyệt lần lượt các nhóm có trong danh sách nhóm, kiểm tra xem người dùng đã có trong nhóm hay chưa. Trong các nhóm người dùng chưa tham gia, tìm ra nhóm hiện đang có nhiều thành viên nhất.

Bài toán này có thể được thực hiện như sau:

```
1 def member(ten_fb,group_fb,group):#tính số thành viên trong group
2     kq = 0
3     for i in range(len(ten_fb)):
4         if group in group_fb[i]:
```

```
5               kq =kq + 1
6       return kq
7
8 input_file = open("group_fb.inp", encoding="utf8")
9 ten_fb=[]
10 group_fb=[]
11 for line in input_file.readlines():
12      ten, group=line.split(":")
14      group=group.strip().split(",")
15      for i in range(len(group)):
16          group[i]=group[i].strip()
17      group_fb.append(group)
18 all_group =[] # Danh sách các nhóm
19 for i in range(0,len(group_fb)):
20      for j in range(0,len(group_fb[i])):
21          if group_fb[i][j] not in all_group:
22              all_group.append(group_fb[i][j])
23
24 input_name = input("Nhập vào tên  tài khoản facebook cần gợi ý:")
25 index = -1
26 for i in range(0,len(ten_fb)): # tìm vị trí của tài khoản trong danh sách
27     if input_name==ten_fb[i]:
28         index = i
29 if (index==-1):
30      print("Không tồn tại tài khoản cần tìm trong danh sách")
31 else:
32      max_member = -1
33      group_result = all_group[0]
34      for i in range(0,len(all_group)):
35           if all_group[i] not in group_fb[index]: # chưa tham gia nhóm này
36              if  member(ten_fb,group_fb,all_group[i])>max_member:
37                  max_member = member(ten_fb,group_fb,all_group[i])
38                  group_result = all_group[i]
39      if (max_member!=-1):
40          print("Nhóm đề xuất tham gia là : ",group_result)
41          print("Số thành viên hiện tại trong nhóm là:",max_member )
```

2. Bài toán này có thể được thực hiện như sau:

```
1 def nhomchung(A,B): # tính số phần tử chung có trong 2 danh sách
2      kq = 0
3      for i in range(0,len(A)):
```

```
4          if A[i] in B:
5              kq= kq+1
6      return kq
7 input_file = open("group_fb.inp", encoding="utf8")
8 ten_fb=[]
9 group_fb=[]
10 for line in input_file.readlines():
11     ten, group=line.split(":")
12     ten_fb.append(ten)
13     group=group.strip().split(",")
14     for i in range(len(group)):
15         group[i]=group[i].strip()
16     group_fb.append(group)
17 max_group = -1
18 names_fb = (ten_fb[0],ten_fb[0])
19 for i in range(0,len(ten_fb)):
20     for j in range(0,len(ten_fb)):
21         if i!=j:
22             if nhomchung(group_fb[i],group_fb[j])>max_group:
23                 max_group = nhomchung(group_fb[i],group_fb[j])
24                 names_fb = (ten_fb[i],ten_fb[j])
25 if max_group!= -1:
26   print("Hai người có số nhóm chung lớn nhất là:",names_fb,"số nhóm chung
là:",max_group)
```

Bài toán tìm hai bạn có số bạn chung lớn nhất được thực hiện tương tự.

BÀI 13. KĨ THUẬT DUYỆT QUAY LUI

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Biết và trình bày được ý tưởng của kĩ thuật quay lui thông qua một số ví dụ đơn giản.

– Nhận ra được mối liên quan giữa thiết kế thuật toán theo kĩ thuật quay lui và kĩ thuật đệ quy.

– Thực hiện được chương trình sinh tất cả các xâu nhị phân bằng kĩ thuật quay lui.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

GV: Có thể chuẩn bị các phương tiện sau:

– Sơ đồ một trò chơi mê cung trên giấy để HS có thể tự chơi bằng cách vẽ đường đi trên mê cung hoặc di chuyển nhân vật trên sơ đồ.

– Phần mềm mô phỏng trò chơi mê cung cho phép HS có thể tự chơi trên máy tính.

HS: Vở ghi và kiến thức Python.

C GỢI Ý PHƯƠNG PHÁP HƯỚNG DẪN THỰC HÀNH

– GV giới thiệu tổng quan mục đích của bài thực hành, các nhiệm vụ cần thực hiện.

– Với mỗi nhiệm vụ, GV đọc đề bài, phân tích cách làm, sau đó yêu cầu HS thực hiện dựa trên SGK đã có.

– Trong quá trình thực hiện nếu HS gặp khó khăn, GV sẽ hướng dẫn và giải quyết các khó khăn đó của HS.

D GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động 1	Hoạt động 2	Hoạt động 3
Tìm hiểu ý tưởng của thuật toán quay lui	Tìm hiểu mô hình tổng quát của kĩ thuật duyệt quay lui	Thiết kế chương trình sinh tất cả các dãy nhị phân

Hoạt động khởi động

Mục đích: HS làm quen với bài toán mê cung cùng lịch sử hình thành thuật toán vét cạn quay lui. Thông qua trò chơi HS sẽ thấy hấp dẫn hơn và muốn khám phá kĩ thuật lập trình quay lui.

GV có thể tổ chức cho các nhóm HS chơi thử trò chơi mê cung trên sơ đồ giấy hoặc phần mềm mô phỏng trò chơi này.

GV hỏi và cho phép HS trình bày để xuất các phương án giải bài toán mê cung này.

1. KĨ THUẬT DUYỆT QUAY LUI

Hoạt động 1. Tìm hiểu ý tưởng của thuật toán quay lui

Mục đích. HS biết được ý tưởng ban đầu của thuật toán quay lui dựa trên mô hình bài toán mê cung.

Chú ý: Với hoạt động này HS chưa cần hiểu rõ các ý tưởng lập trình thực sự của thuật toán quay lui.

Tổ chức hoạt động:

Từ mô hình trò chơi mê cung đã giới thiệu trong phần khởi động, GV sẽ yêu cầu HS thử phác thảo và trao đổi về ý tưởng của thuật toán tìm đường đi bằng vét cạn, quay lui với bài toán mê cung. GV nhấn mạnh rằng chỉ là tìm và mô tả ý tưởng, chứ chưa là lời giải chính xác khoa học của thuật toán này.

Sau đó, GV đưa ra sơ đồ mã giả của thuật toán này và yêu cầu các nhóm trao đổi, thảo luận để hiểu đoạn mã giả này. GV có thể đặt các câu hỏi khác nhau để HS trả lời, thông qua đó hiểu hơn ý nghĩa của ý tưởng này.

Cuối cùng, GV chốt kiến thức.

Hoạt động củng cố kiến thức

1. Có. Khi thực hiện xong vòng lặp tại dòng 5 thì hàm sẽ dừng. Có nghĩa là từ vị trí hiện thời sẽ không thể đi tiếp được nữa, cần "quay lui".

2. Lệnh gọi hàm chính là "Tìm bước đi tiếp theo".

3. Lệnh này đã có chính là dòng lệnh 2.

2. MÔ HÌNH TỔNG QUÁT CỦA THUẬT TOÁN DUYỆT QUAY LUI

Hoạt động 2. Tìm hiểu mô hình tổng quát của kĩ thuật duyệt quay lui

Mục đích. HS biết được mô hình tổng quát của kĩ thuật quay lui được mô tả bằng đệ quy.

Tổ chức hoạt động: GV mô tả mô hình tổng quát của kĩ thuật duyệt quay lui như đã được trình bày trong SGK.

Nghiệm của bài toán sẽ được biểu diễn bằng dãy các số nguyên:

$$(x_1, x_2, ..., x_n)$$

Việc tìm kiếm sẽ mở rộng bắt đầu từ dãy rỗng và từ trạng thái hiện thời $(x_1, x_2, ..., x_k)$ mở rộng lên thành $(x_1, x_2, ..., x_k, x_{k+1})$, nếu không tìm thấy x_{k+1} thì quay lui lại để tìm phần tử x_k tiếp theo, nếu không tìm thấy thì lại quay lui,... Cứ như vậy cho đến khi tìm thấy nghiệm hoặc khi kết thúc thì thông báo vô nghiệm.

Chú ý: Đây là phần kiến thức khó nhất của bài học cũng như cả cuốn sách chuyên đề này. Không yêu cầu HS hiểu hoàn toàn. Sau này thông qua các ví dụ, bài tập, HS sẽ hiểu hơn.

Hoạt động củng cố kiến thức

1. Trạng thái "quay lui" của thuật toán nằm ở sau dòng lệnh 6, khi kết thúc Backtracking() thì cần "quay lui" để quay lại vòng lặp tiếp theo của lệnh for, tại dòng 5.

2. Có. Cần bổ sung một biến tổng thể, ví dụ count, biến này sẽ được tự động tăng lên 1 đơn vị sau lệnh 3, bên trong lệnh if.

3. BÀI TOÁN SINH XÂU NHỊ PHÂN

Hoạt động 3. Thiết kế chương trình sinh tất cả các dãy nhị phân

Mục đích. HS hiểu và thực hiện được chương trình sinh các dãy nhị phân bằng kĩ thuật duyệt quay lui.

Tổ chức hoạt động: GV mô tả bài toán sinh các dãy nhị phân độ dài n bằng kĩ thuật quay lui dựa trên sơ đồ thuật toán tổng quát đã nêu trong Hoạt động 2. GV chú ý các kí hiệu của bài toán này tương ứng với kí hiệu của thuật toán tổng quát như sau:

Bài toán sinh dãy nhị phân độ dài n	**Thuật toán quay lui tổng quát**
$A = [a_0, a_1, ..., a_{n-1}]$	Không gian cần tìm nghiệm $(x_1, x_2, ..., x_k)$
$S_k = [0, 1]$	S_k = miền các giá trị x_k có thể tìm tiếp theo bộ $(x_1, x_2, ..., x_{k-1})$
$D_k = [0, 1]$	D_k = miền giá trị của x_k.

GV giao cho các nhóm làm các chương trình 1 và 2, so sánh kết quả, sự giống nhau và khác nhau của hai chương trình.

Hoạt động củng cố kiến thức

1. Trong chương trình 1, thao tác quay lui sẽ thực hiện ngay sau khi kết thúc lệnh **genBinary(A,k+1)** tại dòng 7.

2. Lệnh **A.pop()** tại dòng 8 chính là thao tác "quay lui" của thuật toán. Lệnh này sẽ xoá đi phần tử cuối của dãy A mà phần tử này mới được thêm vào tại dòng lệnh 6.

Lệnh này không có trong chương trình 1 vì hai chương trình 1 và 2 thực hiện việc tạo dãy A theo hai cách khác nhau. Chương trình 1 tạo sẵn dãy A có độ dài n và sau mỗi bước sẽ bổ sung phần tử bằng cách gán trực tiếp, do đó chương trình 1 không cần lệnh **A.pop()**. Chương trình 2 sẽ thiết lập dãy A từ dãy rỗng, mỗi lần bổ sung phần tử (đi tiếp) sẽ dùng lệnh **A. append()**, do đó khi quay lui sẽ cần dùng lệnh **A. pop()**.

Hoạt động luyện tập

1. Chương trình sau sinh tất cả các xâu nhị phân độ dài n và cuối cùng in tổng số nghiệm tìm được.

```
1 def genBinary(A,k):
2     global count
3     if k == n:
4         print(A)
5         count = count + 1
6     else:
7         for i in range(2):
8             A.append(i)
9             genBinary(A,k + 1)
```

```
10              A.pop()
11
12      n = 3
13      count = 0
14      genBinary([],0)
15      print("Có tất cả",count,"nghiệm")
```

2. Chương trình có thể như sau:

```
1 Color = ["R","G","B"]
2 def genThreenary(A,k):
3     if k == n:
4         print(A)
5     else:
6         for i in range(3):
7         A.append(Color[i])
8         genThreenary(A,k + 1)
9         A.pop()
10
11 n = 2
12 genThreenary([],0)
```

Hoạt động vận dụng

1. Chương trình có thể như sau:

```
1 Hex = ["0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F"]
2 def genThreenary(A,k):
3     if k == n:
4         print("".join(A))
5     else:
6         for i in range(16):
7             A.append(Hex[i])
8             genThreenary(A,k + 1)
9             A.pop()
10 n = 3
11 genThreenary([],0)
```

2. Chương trình có thể như sau:

```
1 Bin = ["0","1"]
2 def genBinary(A,k):
3    if k == n:
4        print("".join(A))
5    else:
6        for i in range(2):
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

```
7               A.append(Bin[i])
8               genBinary(A,k + 1)
9               A.pop()
10
11 n = 3
12 genBinary([],0)
```

BÀI 14. THỰC HÀNH KĨ THUẬT DUYỆT QUAY LUI

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Biết cách sửa dụng kĩ thuật quay lui giải quyết một số bài toán phù hợp với kĩ thuật này và cài đặt thuật toán.

– Nhận ra các vấn đề có thể giải quyết bằng kĩ thuật quay lui và viết mã nguồn cho kĩ thuật quay lui để giải quyết các vấn đề đó.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển được năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

GV: Máy tính đã cài đặt Python và máy chiếu; Hình ảnh, sơ đồ minh hoạ cho các bước thực hiện trên một mẫu dữ liệu đơn giản hoặc có thể sử dụng các phần mềm mô phỏng thuật toán để minh hoạ thêm trong quá trình giảng dạy.

HS: Vở và kiến thức Python cơ bản.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động khởi động

Mục đích: HS nắm được dạng bài toán có thể giải bằng kĩ thuật duyệt quay lui đó là các bài toán tìm nghiệm trong không gian tìm kiếm dạng cây ví dụ như các bài toán tìm đường (đường đi trong mê cung, biểu đồ Hamilton, đường đi quân mã trong bàn cờ), xếp N quân hậu, sinh ra các tổ hợp.

Hoạt động thực hành: Nhiệm vụ 1

Mục đích: giúp HS củng cố kĩ năng dùng kĩ thuật quay lui.

Tổ chức hoạt động: GV có thể để HS luyện tập cá nhân hoặc nhóm 2-3 em. Với các HS gặp khó khăn khi cài đặt thuật toán, GV gợi ý HS dựa trên mã nguồn bài toán sinh chuỗi nhị phân ở bài học trước, chỉ cần thay dữ liệu từ điển 2 bit nhị phân bằng 4 phần tử A, T, G, X.

Hoạt động thực hành: Nhiệm vụ 2

Mục đích: giúp học sinh luyện tập kĩ thuật quay lui để sinh ra các tổ hợp trong đó số lượng các phần tử chưa xác định ở thời điểm viết chương trình.

Tổ chức hoạt động: GV nên để HS luyện tập cá nhân. GV phân tích thêm vì đây chỉ là một bài luyện tập nên chưa chú ý đến ngữ nghĩa và ngữ pháp của các câu văn được sinh ra.

Hoạt động luyện tập

1. Để in ra kết quả chỉ gồm các xâu kí tự "A", "T", "G","X", chúng ta cần sửa đoạn in kết quả tại dòng 4 của chương trình 1. Dưới đây là một cách:

```
1 #Định nghĩa hàm đệ quy thêm các nucleotit vào chuỗi ADN
2 def genDNASection(A,k):
3     if k == n:
4         result = ‘’.join(A)  #Sinh kết quả từ danh sách các phần tử
5         print(result)
6     else:
7         for i in range(4):
8             A[k] = dictDNA[i]
9             genDNASection(A,k+1)
10
11 #Nhập độ dài chuỗi DNA, khởi tạo mảng, rồi gọi hàm genDNASection
12 n = int(input("Hãy nhập độ dài đoạn DNA:"))
13 dictDNA = "ATGX"
14 A = [‘ ’]*n
15 genDNASection(A,0)
```

2. Trong Nhiệm vụ 2, động tác quay lui nằm ở dòng 11 vì dòng 11 nhằm khôi phục trạng thái danh sách các từ đã được lựa chọn (biến words) trước khi thực hiện dòng 9, sau đó thử nghiệm bằng việc thêm một từ khác chưa có trong danh sách từ đã lựa chọn.

Hoạt động vận dụng

1. Bài tập này giúp học sinh củng cố cách cài đặt kĩ thuật duyệt quay lui. HS có thể dựa trên mã nguồn của Nhiệm vụ 1 hoặc 2 để viết chương trình với lưu ý là số lượng phần tử để tính hoán vị được nhập động bởi người dùng. Mã nguồn tham khảo như sau:

```
def genPermutations(permutation, elements, k):
    if k == len(elements):
        print(permutation)
    else:
        for number in elements:
            if not number in permutation:
                permutation.append(number)
                genPermutations(permutation, elements, k+1)
                permutation.pop()

dictionary = []
n = int(input("Hãy nhập số các phần tử n: "))
for i in range(n):
    dictionary.append(i+1)
permutation = []
print("Các hoán vị từ các số từ 1 đến",n,"là:")
genPermutations(permutation, dictionary, 0)
```

2. Bài tập sinh đa thức bậc k giúp học sinh luyện tập cách sinh ra các hoán vị bằng kĩ thuật duyệt quay lui. Đầu tiên chương trình cần yêu cầu người dùng nhập vào k+1 hệ số. Bài toán được giải quyết nếu coi một đa thức là một danh sách các hệ số và các đa thức khác nhau được sinh ra bằng cách hoán vị các hệ số từ một đa thức ban đầu. Lưu ý người dùng có thể nhập các hệ số là các giá trị trùng nhau nhưng chương trình cần tránh sinh ra các danh sách hệ số giống nhau và hệ số đầu tiên cần phải khác 0 để đảm bảo đây là một đa thức bậc k.

Chương trình như sau:

```
def genPolynomials(coefs, k):
    if k == len(coefs) and coefs[0]!= "0":#in ra đa thức nếu hệ số đầu khác 0
        for i in range(len(coefs)):
            if coefs[i]== "0": #Phần tử với hệ số =0 không cần in
                continue
            if i!=0 and float(coefs[i]) > 0: #In dấu + trước các hệ số dương
                print("+", sep='', end = '')
            print(coefs[i], sep='', end = '') #in hệ số
            if len(coefs)-i-1 >=2: #In x và số mũ nếu số mũ >= 2
                print("x^", len(coefs)-i-1, sep='', end = '')
            elif len(coefs)-i-1 ==1:#Chỉ in x nếu số mũ = 1
                print("x", sep='', end = '')
        print()
    else:
        for i in range(k, len(coefs)):
            if i == k: #Nếu đang xét phần tử đầu tiên trong phần còn lại
```

```
                genPolynomials(coefs, k+1) #Tiếp tục thêm các hệ số
            elif coefs[i] != coefs[k]: #Nếu hai phần tử khác nhau
                coefs[i], coefs[k] = coefs[k], coefs[i] #Tráo hai hệ số
                        genPolynomials(coefs, k+1) #Tiếp tục thêm các hệ số
                coefs[i], coefs[k] = coefs[k], coefs[i] #Tráo lại hai hệ số

coefs = input("Hãy nhập các hệ số phân cách bởi dấu cách: ").split()#coefs
    chứa danh sách các hệ số
genPolynomials(coefs, 0)
```

BÀI 15. BÀI TOÁN XẾP HẬU

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Biết và thực hiện được chương trình giải bài toán xếp Hậu trên bàn cờ.

– Thực hiện được lập trình cho bài toán xếp hậu theo kĩ thuật quay lui.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông; sử dụng được ngôn ngữ lập trình bậc cao, qua đó phát triển tư duy điều khiển và tự động hoá;

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

– *GV:* Có thể dùng bàn cờ thật làm dụng cụ hỗ trợ cho bài học này. Cũng có thể dùng hình ảnh hoặc phần mềm mô phỏng chơi cờ để mô phỏng bài toán.

– *HS*: Vở ghi và kiến thức Python cơ bản, kiến thức về kĩ thuật quay lui.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động 1	Hoạt động 2
Tìm hiểu mô hình bài toán xếp Hậu trên bàn cờ vua tổng quát.	Tìm hiểu thuật toán và cách giải bài toán xếp Hậu trên bàn cờ vua.

Hoạt động khởi động

Mục đích: HS làm quen với bài toán rất nổi tiếng xếp 8 quân Hậu trên bàn cờ. Bài toán có thể mở rộng thành: Tìm tất cả các phương án xếp n quân Hậu trên bàn cờ vuông bậc n sao cho không quân nào khống chế quân nào.

GV cho các nhóm HS, mỗi nhóm tìm một phương án để tạo sự tò mò muốn biết cách giải bài toán này trên máy tính hoặc GV có thể cho các nhóm tìm tất cả các phương án cho trường hợp n = 4, từ đó HS sẽ hình dung được công việc cần phải làm là gì.

1. MÔ TẢ BÀI TOÁN XẾP HẬU TRÊN BÀN CỜ VUA

Hoạt động 1. Tìm hiểu mô hình bài toán xếp Hậu trên bàn cờ vua tổng quát

Mục đích. HS biết được mô hình dữ liệu chính thức của bài toán từ đó có thể thiết lập được thuật toán giải bằng quay lui.

Tổ chức hoạt động:

– Đầu tiên GV cho HS quan sát bàn cờ thực tế để HS tự thiết lập dữ liệu cho bài toán. GV cho nhận xét, đánh giá và chốt lại phương án chính thức sẽ được dùng.

– GV giới thiệu và cùng HS thiết lập được mô hình nghiệm của bài toán dựa trên cấu trúc dữ liệu đã thiết lập, từ đó sẽ dễ dàng thấy cách tiếp cận vét cạn quay lui của bài toán.

– Điểm quan trọng nhất của hoạt động này là xác định điều kiện xếp được quân Hậu tiếp theo tại cột k sau khi đã xếp xong tại các cột bên trái. Mệnh đề mô tả điều kiện của quân cờ tại cột k chính là điều kiện tính S_k của sơ đồ thuật toán quay lui tổng quát đã học trong Bài 13.

– Cuối cùng, GV chốt kiến thức.

Hoạt động củng cố kiến thức

1. Với n = 4, A[0] = 2, A[1] = 0, thì A[2] = 3 (chỉ có 1 giá trị duy nhất).

2. Với n = 5, A[0] = 0, A[1] = 3, thì A[2] = 1 (chỉ có 1 giá trị duy nhất).

2. THIẾT LẬP LỜI GIẢI BÀI TOÁN XẾP HẬU TỔNG QUÁT

Hoạt động 2. Tìm hiểu thuật toán và cách giải bài toán xếp Hậu trên bàn cờ vua

Mục đích. HS biết và hiểu được hoạt động của chương trình giải bài toán xếp Hậu trên bàn cờ tổng quát.

Tổ chức hoạt động:

– GV mô tả hàm chính **trynext()** của thuật toán.

– Sau đó, GV giao cho các nhóm cùng hoàn thiện chi tiết thuật toán này bằng chương trình.

– Cuối cùng, GV đưa toàn bộ lời giải và chương trình hoàn chỉnh để các nhóm quan sát và hiểu chương trình. Cho các nhóm lần lượt chạy chương trình với các dữ liệu n khác nhau.

Hoạt động củng cố kiến thức

1. Với n = 3 bài toán không có nghiệm.

2. Cần khai báo biến *ncount* trong hàm **trynext()** với từ khóa *global* để đảm bảo biến *ncount* thực sự là biến tổng thể và có tác dụng trong toàn bộ chương trình và bên trong hàm **trynext()**.

Hoạt động luyện tập

1. Với n = 4, bài toán có hai nghiệm sau:

Nghiệm 1:

```
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0
```

Nghiệm 2:

```
0 1 0 0
0 0 0 1
1 0 0 0
0 0 1 0
```

2. Nếu mô phỏng lưới ô vuông đánh số từ dưới lên trên thì chương trình trên vẫn đúng về lôgic chỉ cần sửa lại lệnh in kết quả ra màn hình. Vậy chỉ cần sửa hàm **showQueen()**.

Sửa lại hàm này như sau:

```
def showQueen(A):
    n = len(A)
    for i in range(n-1,-1,-1):
        for j in range(n):
            if A[j] == i:
                print(1, end = " ")
            else:
                print(0, end = " ")
        print()
```

Hoạt động vận dụng

1. Các giá trị Q(n) được tính trong bảng sau:

n	1	2	3	4	5	6	7	8	9	10
Q(n)	1	0	0	2	10	4	40	92	352	724

2. Các giá trị Q(n) với n = 11, 12, 13 có thể tính được như sau (sẽ tốn khá nhiều thời gian)

n	11	12	13
Q(n)	2680	14200	73712

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

BÀI 16. THỰC HÀNH THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT DUYỆT QUAY LUI

A MỤC TIÊU

1. KIẾN THỨC, KĨ NĂNG

– Ôn lại kĩ thuật duyệt quay lui để thiết kế thuật toán cho các bài toán tìm kiếm.

– Thực hành kĩ thuật duyệt quay lui với một số bài toán phức tạp hơn.

– Cài đặt kĩ thuật duyệt quay lui cho các bài toán tìm kiếm phức tạp.

2. NĂNG LỰC, PHẨM CHẤT

– Hình thành, phát triển được năng lực giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông.

– Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ.

B CHUẨN BỊ

– *GV:* Có thể dùng bàn cờ thật làm dụng cụ hỗ trợ cho bài học này. Cũng có thể dùng hình ảnh hoặc phần mềm mô phỏng chơi cờ để mô phỏng bài toán.

– *HS*: Vở ghi và kiến thức Python cơ bản, kiến thức về kĩ thuật quay lui.

C GỢI Ý CÁC HOẠT ĐỘNG DẠY HỌC CHÍNH

Hoạt động khởi động

Mục đích: HS suy nghĩ về việc áp dụng kĩ thuật duyệt quay lui cho bài toán tìm đường mê cung, và vấn đề sinh ra các nhánh tìm kiếm tại một vị trí của cây tìm kiếm trong bài toán này. Các bước đi tiếp theo là tất cả các hướng từ vị trí đó, cụ thể là hướng lên xuống trái phải trong bài toán mê cung này.

Hoạt động thực hành: Giải bài toán mê cung

Tổ chức hoạt động: GV có thể để HS luyện tập cá nhân hoặc nhóm 2-3 em. HS trước hết cần hiểu cấu trúc dữ liệu lưu trữ cấu trúc mê cung và vết đường đi được tìm thấy. Sau đó HS cần hiểu ý tưởng thuật toán, triển khai ý tưởng bằng mã nguồn Python.

Hoạt động luyện tập

1. Để đặt vị trí xuất phát tại ô giữa của mê cung cần chỉnh giá trị tham số x, y khi gọi hàm solveMaze, cụ thể tại dòng 47 sửa thành:

```
if solveMaze (maze, m, n, m//2, n//2, sol) == False:
```

Vị trí thoát của mê cung được thiết lập ở dòng 21. Theo yêu cầu vị trí thoát có thể là một trong hai trường hợp trái trên hoặc phải dưới thì chúng ta có thể sửa như sau:

```
if maze[y][x] == 0 and ((x == n - 1 and y == m - 1) or (x ==0 and y ==0)):
```

2. Để đánh dấu "x" vào các ô là tường, có thể thêm một đoạn mã trước khi in ra ma trận sol để duyệt qua các ô của maze, nếu là tường thì sửa ô tương ứng của sol thành x. Tham khảo mã nguồn sau trong đó đoạn sửa ma trận sol ở dòng 50 đến 53.

```
1 # Hàm readmaze/writemaze để đọc/ghi mê cung
2 def readmaze():
3     file = open("maze.inp")
4     maze = []
5     for row in file:
6         h = [int(x) for x in row.split()]
7         maze.append(h)
8     file.close()
9     return maze
10 def writemaze(outmaze):
11     f = open("maze.out","w")
12     for i in outmaze:
13         for j in i:
14             print(j,file = f,end =" ")
15         print(file = f)
16     f.close()
17
18 # Hàm đệ quy tìm đường đi trong mê cung theo phương pháp quay lui
19 def solveMaze(maze,  m, n, x, y, sol):
20     #Nếu đã đến ô bên phải phía dưới thì đã tìm được đường và kết thúc
21     if x == n - 1 and y == m - 1 and maze[y][x] == 0:
22         sol[y][x] = 1
23         writemaze(sol)
24         return True
25      #Nếu ô x,y này rỗng và chưa từng được đi qua thì tìm đường đi tiếp
26       if x >= 0 and x < n and y >= 0 and y < m and maze[y][x] == 0 and
        sol[y][x] == 0:
27           if sol[y][x] == 1:
28               return True
29           if solveMaze(maze, m, n, x+1, y, sol):
30               return True
31           if solveMaze(maze, m, n, x, y+1, sol):
32               return True
33           if x> 0 and solveMaze(maze, m, n, x-1, y, sol):
34               return True
```

```
35            if y> 0 and solveMaze(maze, m, n, x, y-1, sol):
36                return True
37            sol[y][x] = 0 #Mọi hướng đều không đi được nên đánh dấu 0
38            return False
39        else:
40            return False
41 #Chương trình chính
42 maze = readmaze()
43 m = len(maze)
44 n = len(maze[0])
45 print("Kích thước mê cung:",m,"hàng",n,"cột")
46 sol = [[ 0 for j in range(n) ] for i in range(m)]
47 if solveMaze(maze, m, n, 0, 0, sol) == False:
48     print("Không tồn tại đường đi")
49 else:
50     for i in range(m):
51         for j in range(n):
52             if maze[i][j] == 1:
53                 sol[i][j] = 'x' #Đánh dấu x vào các ô tường
54     writemaze(sol)
55     print("Đường đi khỏi mê cung:")
56     for i in sol:
57     for j in i:
58         print(str(j) + " ", end ="")
59     print()#in kí tự xuống dòng
```

Hoạt động vận dụng

1. Bài này vận dụng cách tìm tất cả các nghiệm bằng kĩ thuật duyệt quay lui. HS có thể sửa mã nguồn trong phần nhiệm vụ bằng cách luôn duyệt tất cả các đường đi từ vị trí hiện tại, không trả về True dù đã tìm thấy một đường đi (dòng 29-36). Mã nguồn tham khảo như sau:

```
1 def genPermutations(permutation, elements, k):
2 # Hàm readmaze/writemaze để đọc/ghi mê cung
3 def readmaze():
4     file = open("maze-mul.inp")
5     maze = []
6     for row in file:
7         h = [int(x) for x in row.split()]
8         maze.append(h)
9     file.close()
10    return maze
```

```
def printmaze(outmaze):
    for i in outmaze:
        for j in i:
            print(j,end =“ ”)
        print()

# Hàm đệ quy tìm đường đi trong mê cung theo phương pháp quay lui
def solveMaze (maze,  m, n, x, y, sol):
    #Nếu đã đến ô bên phải phía dưới thì đã tìm được đường và kết thúc
    if x == n - 1 and y == m - 1 and maze[y][x] == 0:
        sol[y][x] = 1
        print(“Tìm ra một đường đi:”)
        printmaze(sol)
        return
    #Nếu ô x,y này rỗng thì tìm đường đi tiếp
    if x >= 0 and x < n and y >= 0 and y < m and maze[y][x] == 0:
        if sol[y][x] == 1: #Nếu ô này đã được đi qua thì không thử nữa
            return
        sol[y][x] = 1#Tạm đánh dấu đường ở ô này
        solveMaze(maze, m, n, x+1, y, sol)
        solveMaze(maze, m, n, x, y+1, sol)
        solveMaze(maze, m, n, x-1, y, sol)
        solveMaze(maze, m, n, x, y-1, sol)
        sol[y][x] = 0 #Mọi hướng đều không đi được nên đánh đấu 0
#Chương trình chính
maze = readmaze()
m = len(maze)
n = len(maze[0])
print(“Kích thước mê cung:”,m,“hàng”,n,“cột”)
sol = [[ 0 for j in range(n) ] for i in range(m)]
solveMaze(maze, m, n, 0, 0, sol)
```

2. Lời giải bài toán xếp Hậu tổng quát được cải tiến từ Hoạt động 2 Bài 15 bằng cách sửa đổi hàm *showQueen* để duyệt qua m hàng n cột (bàn cờ không còn vuông nữa). Mã nguồn tham khảo như sau:

```
def showQueen(A):
    for i in range(m):
        for j in range(n):
            if A[j] == i:
                print(1, end = “ ”)
            else:
                print(0, end = “ “)
```

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

```
        print()

def isOK(i,j):
    for k in range(j):
        if A[k] == i:
            return False
        if abs(A[k] - i) == abs(j - k):
            return False
     return True

def trynext(k):
    global solutionCount
    if k == n:
        solutionCount = solutionCount + 1
        print("Phương án",solutionCount,":")
        showQueen(A)
    else:
        for i in range(m):
            if isOK(i,k):
                A[k] = i
                trynext(k+1)

dataIsOK = False
while dataIsOK == False:
    print("Hãy nhập kích thước bàn cờ sao cho hàng nhiều hơn cột")
    m = int (input("Số hàng: «))
    n = int (input("Số cột: "))
    if(m>=n and n>0):
        dataIsOK = True
    else:
        print("Dữ liệu không hợp lệ")

A = [0]*n
solutionCount = 0
trynext(0)
```

3. Bài toán "Mã đi tuần" có thể được giải quyết bằng kĩ thuật duyệt quay lui. Chúng ta có thể hoàn toàn áp dụng mô hình tổng quát của kĩ thuật này được trình bày tại Bài 13. Vấn đề quan trọng nhất là tại mỗi vị trí hiện tại của quân Mã, chúng ta cần tìm ra các nước đi hợp lệ tiếp theo. Nước đi hợp lệ là nước đi để quân Mã vẫn trong bàn cờ và không đi vào ô đã duyệt. Để lưu vết đường đi, chúng ta có thể dùng một mảng tên là board 8×8 (kích thước bàn cờ) để lưu thứ tự nước đi. Board được khởi tạo với tất cả các giá trị -1 thể hiện quân Mã chưa duyệt qua. Lưu ý tại mỗi trạng thái có thể có tối đa 8 bước đi cần thử nghiệm nên không gian cần tìm kiếm có thể rất lớn và thuật toán chạy rất lâu. Sau đây là mã nguồn tham khảo:

```
1 def isOK(x, y, board): #Hàm kiểm tra xem ô x, y có hợp lệ để đi tiếp không
2     if(x >= 0 and y >= 0 and x < n and y < n and board[x][y] == -1):
3         return True
4     return False
5
6 def printSolution(n, board): # Hàm in giải pháp với thứ tự các nước đi trong
bàn cờ
7     for i in range(n):
8         for j in range(n):
9             print(board[i][j], end=› ‹)
10        print()
11
12 def knightTraverse(n, board, curr_x, curr_y, pos):
13     if(pos == n**2): # Đã đi hết toàn bộ bàn cờ
14         return True
15     print(curr_x, curr_y)
16     # Định nghĩa tám hướng đi từ vị trí hiện tại
17     move_x = [2, 1, -1, -2, -2, -1, 1, 2]
18     move_y = [1, 2, 2, 1, -1, -2, -2, -1]
19     for i in range(8):# Thử tất cả 8 hướng đi từ vị trí x,y hiện tại
20         new_x = curr_x + move_x[i]
21         new_y = curr_y + move_y[i]
22         if(isOK(new_x, new_y, board)):
23             board[new_x][new_y] = pos # Đánh dấu thứ tự bước đi mới
24             if(knightTraverse(n, board, new_x, new_y, pos+1)):
25                 return True
26             board[new_x][new_y] = -1 # Quay lui đánh dấu ô này không đi được
27     return False
28
29 n = 8# Kích thước bàn cờ
30 board = [[-1 for i in range(n)]for i in range(n)]# Khởi tạo bàn cờ
31
32 x = int(input("Hãy nhập vị trí hàng (0-7):"))
33 y = int(input("Hãy nhập vị trí cột (0-7):"))
34 board[x][y] = 0 #Đánh dấu vị trí đầu tiên của quân Mã trên bàn cờ
35
36 if(not knightTraverse(n, board, x, y, 1)):
37     print("Không có đường đi hợp lệ")
38 else:
39     printSolution(n, board)
```

Nhà xuất bản Giáo dục Việt Nam xin trân trọng cảm ơn
các tác giả có tác phẩm, tư liệu được sử dụng, trích dẫn trong cuốn sách này.

Chịu trách nhiệm xuất bản:
Chủ tịch Hội đồng thành viên – Tổng Giám đốc NGUYỄN TIẾN THANH

Chịu trách nhiệm nội dung:
Tổng biên tập PHẠM VĨNH THÁI

Biên tập nội dung: PHẠM THỊ THANH NAM – NGUYỄN THỊ THANH XUÂN – VŨ THỊ VÂN
Thiết kế sách: TRẦN NGỌC LÊ
Trình bày bìa: NGUYỄN BÍCH LA
Sửa bản in: PHẠM THỊ TÌNH – TRẦN THU HÀ
Chế bản: PHÒNG CHẾ BẢN, CÔNG TY CP DVXB GIÁO DỤC HÀ NỘI

Bản quyền thuộc Nhà xuất bản Giáo dục Việt Nam.
Tất cả các phần của nội dung cuốn sách này đều không được sao chép, lưu trữ, chuyển thể dưới bất kì hình thức nào khi chưa có sự cho phép bằng văn bản của Nhà xuất bản Giáo dục Việt Nam.

CHUYÊN ĐỀ HỌC TẬP TIN HỌC 11
ĐỊNH HƯỚNG KHOA HỌC MÁY TÍNH (SÁCH GIÁO VIÊN)

Mã số:

In ..... cuốn (QĐ ..... SLK), khổ 19 x 26,5cm.
In tại Công ty cổ phần in .....
Số ĐKXB: ...../CXBIPH/...../GD
Số QĐXB: ..... / QĐ-GD ngày ... tháng ... năm ....
In xong và nộp lưu chiểu tháng ..... năm .....
Mã số ISBN:



HUÂN CHƯƠNG HỒ CHÍ MINH

BỘ SÁCH GIÁO VIÊN LỚP 11 – KẾT NỐI TRI THỨC VỚI CUỘC SỐNG

1. Ngữ văn 11, tập một – SGV
2. Ngữ văn 11, tập hai – SGV
3. Chuyên đề học tập Ngữ văn 11 – SGV
4. Toán 11 – SGV
5. Chuyên đề học tập Toán 11 – SGV
6. Lịch sử 11 – SGV
7. Chuyên đề học tập Lịch sử 11 – SGV
8. Địa lí 11 – SGV
9. Chuyên đề học tập Địa lí 11 – SGV
10. Giáo dục Kinh tế và Pháp luật 11 – SGV
11. Chuyên đề học tập Giáo dục Kinh tế và Pháp luật 11 – SGV
12. Vật lí 11 – SGV
13. Chuyên đề học tập Vật lí 11 – SGV
14. Hoá học 11 – SGV
15. Chuyên đề học tập Hoá học 11 – SGV
16. Sinh học 11 – SGV
17. Chuyên đề học tập Sinh học 11 – SGV
18. Công nghệ 11 – Công nghệ cơ khí – SGV
19. Chuyên đề học tập Công nghệ 11 – Công nghệ cơ khí – SGV
20. Công nghệ 11 – Công nghệ chăn nuôi – SGV
21. Chuyên đề học tập Công nghệ 11 – Công nghệ chăn nuôi – SGV
22. Tin học 11 – SGV
23. Chuyên đề học tập Tin học 11 – Định hướng Tin học ứng dụng – SGV
24. Chuyên đề học tập Tin học 11 – Định hướng Khoa học máy tính – SGV
25. Mĩ thuật 11 – SGV
26. Chuyên đề học tập Mĩ thuật 11 – SGV
27. Âm nhạc 11 – SGV
28. Chuyên đề học tập Âm nhạc 11 – SGV
29. Hoạt động trải nghiệm, hướng nghiệp 11 – SGV
30. Giáo dục thể chất 11 – Bóng chuyền – SGV
31. Giáo dục thể chất 11 – Bóng đá – SGV
32. Giáo dục thể chất 11 – Cầu lông – SGV
33. Giáo dục thể chất 11 – Bóng rổ – SGV
34. Giáo dục quốc phòng và an ninh 11 – SGV
35. Tiếng Anh 11 – Global Success – SGV

Website: https://www.nxbgd.vn

KẾT NỐI TRI THỨC VỚI CUỘC SỐNG