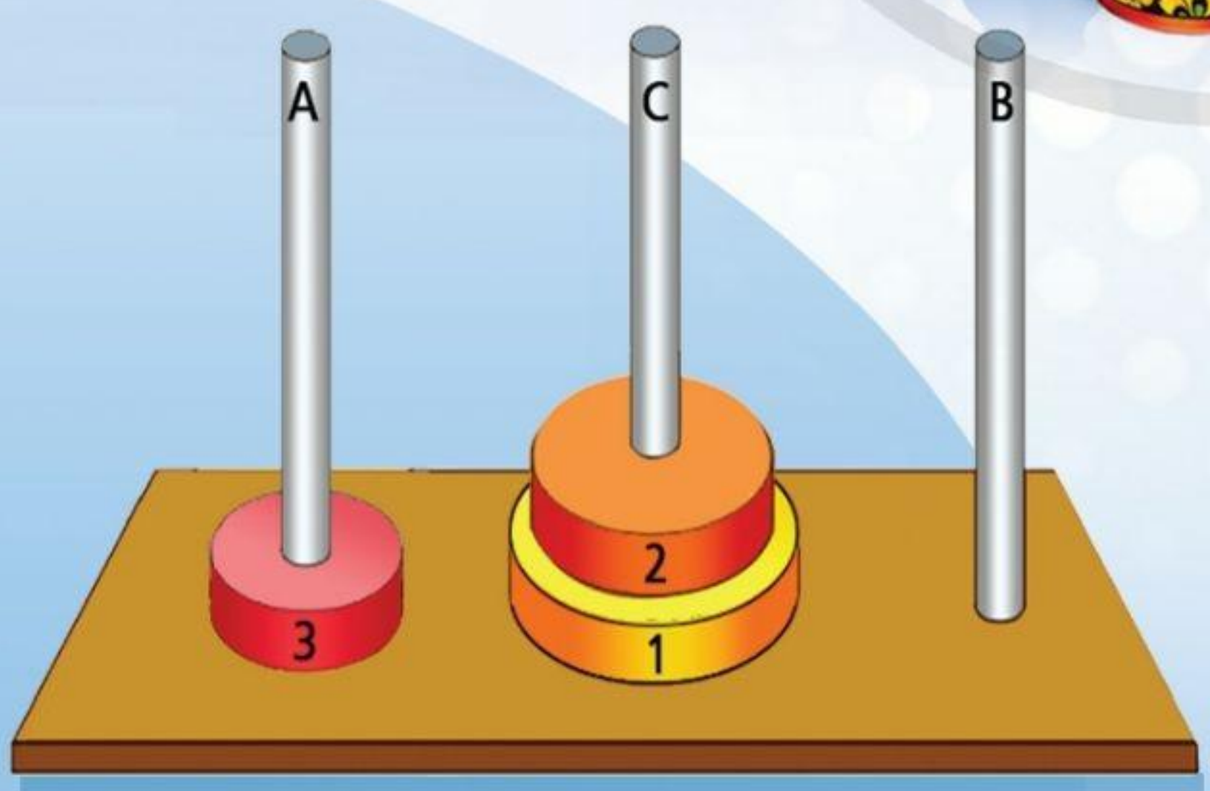
HỒ SĨ ĐÀM (Tổng Chủ biên) – ĐỖ ĐỨC ĐÔNG (Chủ biên)
NGUYỄN KHÁNH PHƯƠNG – ĐỖ PHAN THUẬN

CHUYÊN ĐỀ HỌC TẬP
Tin học
KHOA HỌC MÁY TÍNH
11

BẢN MẪU



NHÀ XUẤT BẢN ĐẠI HỌC SƯ PHẠM
CÔNG TY CỔ PHẦN ĐẦU TƯ XUẤT BẢN - THIẾT BỊ GIÁO DỤC VIỆT NAM

Đọc bản mới nhất trên hoc10.vn
Bản sách mẫu

HỒ SĨ ĐÀM (Tổng Chủ biên) – ĐỖ ĐỨC ĐÔNG (Chủ biên)
NGUYỄN KHÁNH PHƯƠNG – ĐỖ PHAN THUẬN

CHUYÊN ĐỀ HỌC TẬP
Tin học
KHOA HỌC MÁY TÍNH 11

(Sách đã được Bộ trưởng Bộ Giáo dục và Đào tạo phê duyệt sử dụng trong cơ sở giáo dục phổ thông tại Quyết định số 4607/QĐ-BGDĐT ngày 28/12/2022)

NHÀ XUẤT BẢN ĐẠI HỌC SƯ PHẠM

CÔNG TY CỔ PHẦN ĐẦU TƯ
XUẤT BẢN - THIẾT BỊ GIÁO DỤC VIỆT NAM

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

CÁC CHUYÊN ĐỀ

Chuyên đề 1

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

Chuyên đề 3

Thực hành thiết kế thuật toán theo kĩ thuật duyệt

KÍ HIỆU DÙNG TRONG SÁCH

- Khởi động
- Hoạt động
- Luyện tập
- Vận dụng
- Câu hỏi tự kiểm tra

Các em giữ gìn sách cẩn thận, không viết vào sách để sử dụng được lâu dài.

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

LỜI NÓI ĐẦU

Các em học sinh thân mến!

Quyển sách *Chuyên đề học tập Tin học 11 – Khoa học máy tính* sẽ hướng dẫn các em thực hành một số chiến lược thiết kế thuật toán cơ bản. Thông qua đó, giúp các em phát triển tư duy máy tính, năng lực giải quyết vấn đề. Cụm chuyên đề với 35 tiết học gồm ba chuyên đề: *Thực hành thiết kế thuật toán theo kĩ thuật đệ quy*, *Thực hành thiết kế thuật toán theo kĩ thuật chia để trị* và *Thực hành thiết kế thuật toán theo kĩ thuật duyệt*. Trong mỗi chuyên đề, việc thực hành, hoàn thiện sản phẩm đều được chú trọng. Ở cuối Chuyên đề 3, các em được làm việc theo nhóm trong *Dự án học tập: Xây dựng chương trình sử dụng kĩ thuật duyệt*, dự án này sẽ giúp các em khám phá, chủ động tìm hiểu xây dựng chương trình giải quyết các bài toán gần gũi, thú vị sử dụng kĩ thuật duyệt.

Các hoạt động: *Khởi động*; *Hoạt động* kiến tạo kiến thức, kĩ năng; *Luyện tập*; *Vận dụng* và *Câu hỏi tự kiểm tra* trong quyển sách này được thiết kế kĩ lưỡng và logic. Các chuyên đề góp phần giúp các em có nhiều cơ hội hình thành và phát triển năng lực tin học. Các nội dung thực hành giúp các em rèn luyện kĩ năng làm việc nhóm, khả năng khám phá, giải quyết vấn đề và sáng tạo.

Chúc các em hoàn thành tốt các nội dung học tập và tìm được nhiều điều thú vị trong quyển sách này.

Các tác giả

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

MỤC LỤC

Nội dung	Trang
CHUYÊN ĐỀ 1. THỰC HÀNH THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT ĐỆ QUY	5
Bài 1. Khái niệm đệ quy và ví dụ	5
Bài 2. Thuật toán đệ quy	10
Bài 3. Thực hành thiết kế thuật toán đệ quy	16
Bài 4. Thực hành tổng hợp thiết kế thuật toán đệ quy	20
CHUYÊN ĐỀ 2. THỰC HÀNH THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT CHIA ĐỂ TRỊ	24
Bài 1. Ý tưởng chia để trị	24
Bài 2. Kĩ thuật đệ quy trong chia để trị	31
Bài 3. Thực hành ứng dụng thuật toán tìm kiếm nhị phân bằng đệ quy	35
Bài 4. Kĩ thuật chia để trị trong thuật toán sắp xếp trộn	38
Bài 5. Thực hành tổng hợp ứng dụng chia để trị	46
CHUYÊN ĐỀ 3. THỰC HÀNH THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT DUYỆT	49
Bài 1. Kĩ thuật duyệt	49
Bài 2. Kĩ thuật quay lui	55
Bài 3. Thực hành kĩ thuật quay lui	61
Bài 4. Thực hành tổng hợp kĩ thuật duyệt	63
Bài 5. Thực hành kĩ thuật quay lui giải bài toán xếp hậu	65
Bài 6. Dự án: Xây dựng chương trình sử dụng kĩ thuật duyệt	68
Bảng giải thích thuật ngữ	71

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

CHUYÊN ĐỀ 1

THỰC HÀNH THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT ĐỆ QUY

Bài 1

KHÁI NIỆM ĐỆ QUY VÀ VÍ DỤ

Học xong bài này, em sẽ:

- Biết được tính đệ quy vẫn thường xuất hiện trong các sự vật, sự việc ta gặp hằng ngày.
- Nêu được ví dụ cụ thể và mô tả được tính đệ quy trong một số định nghĩa sự vật, sự việc.
- Xác định được phần cơ sở và phần đệ quy trong một mô tả đệ quy.
- Nhận biết được ưu điểm của đệ quy trong mô tả một số đối tượng, thuật toán.

Trong toán học, n giai thừa (kí hiệu $n!$) là tích của n số nguyên dương đầu tiên $n! = n \times (n-1) \times \ldots \times 1$. Vậy ta có thể dùng công thức sau đây để tính $n!$ được không?

$$n! = \begin{cases} n \times (n-1)! \text{ nếu } n \geq 1 \\ 1 \text{ nếu } n = 0 \end{cases}$$

1 Một số ví dụ về đệ quy

Trong toán học có rất nhiều công thức được định nghĩa thông qua chính nó ở phiên bản nhỏ hơn. Những công thức như vậy được gọi là mang tính đệ quy.

Ví dụ 1: Công thức sinh dãy số Fibonacci 0, 1, 1, 2, 3, 5, 8, 13,...

$$F(n) = \begin{cases} F(n-1) + F(n-2) \text{ nếu } n \geq 2 \\ 0 \text{ nếu } n = 0 \\ 1 \text{ nếu } n = 1 \end{cases} \qquad (1)$$

Từ công thức (1) ở trên, ta thấy số Fibonacci thứ n ($n = 0, 1, 2, \ldots$) chính là giá trị của hàm F tại n, được tính bằng tổng giá trị của hàm F tại hai giá trị nhỏ hơn là $n-1$ và $n-2$.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Sử dụng công thức (1) để tiếp tục quá trình tính toán, ta có $F(n-1) = F(n-2) + F(n-3)$, $F(n-2) = F(n-3) + F(n-4)$,... Do đó, nếu cứ gọi đến hàm F như vậy thì việc tính toán sẽ không có điểm dừng nên ta phải bổ sung trường hợp đặc biệt được tính toán sẵn, là hàm F tại $n = 0$ có giá trị 0 và tại $n = 1$ có giá trị 1. Công thức (1) là công thức mang tính đệ quy.

1

Hai công thức sau đều được sử dụng để tính số cách chọn k phần tử từ n phần tử :

$$C(n, k) = \frac{n!}{k!\,(n-k)!} \qquad (2)$$

$$C(n, k) = \begin{cases} C(n-1, k-1) + C(n-1, k) \text{ nếu } 0 < k < n \\ 1 \text{ nếu } k = 0 \\ 1 \text{ nếu } k = n \end{cases} \qquad (3)$$

Theo em, trong hai công thức (2) và (3), công thức nào là công thức mang tính đệ quy? Em hãy giải thích cho lựa chọn của mình.

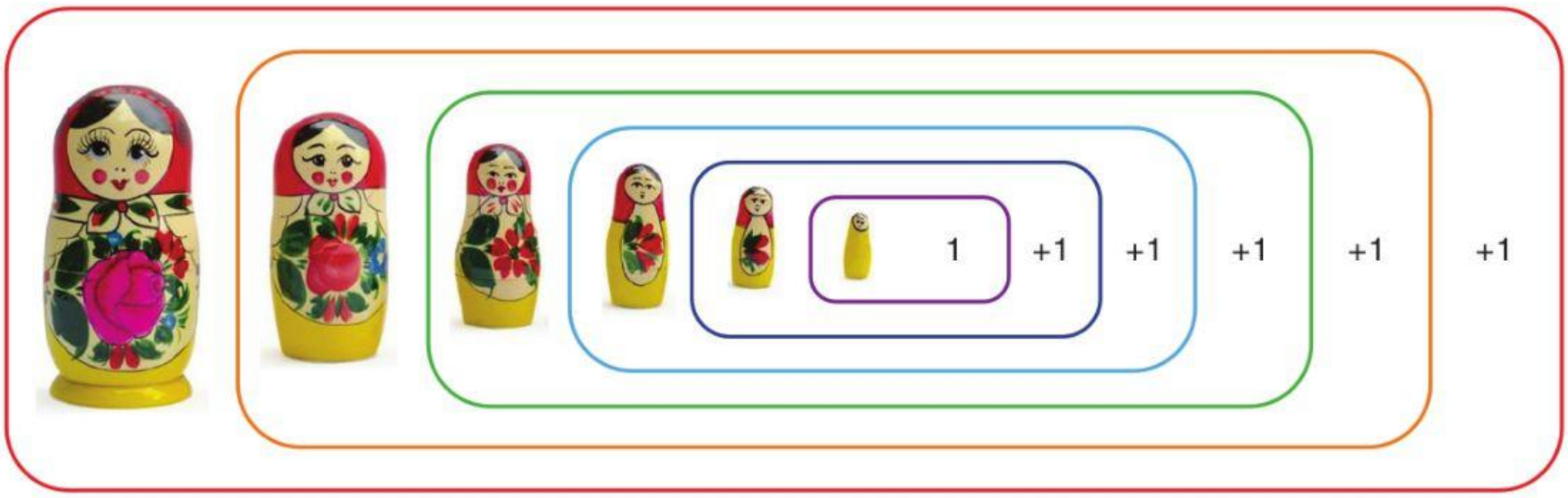
Trong cuộc sống hằng ngày, có rất nhiều sự việc, hiện tượng, cách giải quyết một vấn đề được mô tả dưới góc nhìn mang tính đệ quy. Chúng ta xét ví dụ 2 sau đây.

Ví dụ 2: Bộ búp bê Matryoshka truyền thống của Nga được làm bằng gỗ, gồm nhiều con có kích thước giảm dần, con nhỏ hơn được đặt bên trong con lớn hơn một cách lần lượt (*Hình 1*). Để đếm số lượng búp bê của bộ Matryoshka, ta bắt đầu từ búp bê lớn nhất, số lượng búp bê của bộ bằng 1 cộng với số lượng búp bê có trong búp bê lớn nhất. Tương tự, số búp bê có trong búp bê lớn nhất bằng 1 cộng với số búp bê có trong búp bê con của búp bê lớn nhất.

Áp dụng liên tiếp cách đếm này bằng các thao tác mở búp bê cho đến khi búp bê thu được không có búp bê con nào trong nó. Kết quả thu được là tổng của một dãy các số 1 như minh hoạ ở *Hình 2*.

Hình 1. Bộ búp bê Matryoshka

Đọc bản mới nhất trên hoc10.vn Bản mẫu



Hình 2. Cách đếm số búp bê của bộ Matryoshka

Bài toán đếm số búp bê của bộ Matryoshka được thực hiện nhờ giải liên tục các bài toán đếm đồng dạng với số lượng búp bê giảm dần. Ta nói bài toán này được giải một cách đệ quy. Quá trình đệ quy dừng lại ở búp bê nhỏ nhất trong cùng, tương ứng với bài toán có kích thước nhỏ nhất.

2 Khái niệm và các thành phần cơ bản trong định nghĩa đệ quy

2

Hàm **dem_Bupbe(búp bê A)** ở *Hình 3* được sử dụng để mô tả cách đếm số búp bê của bộ Matryoshka một cách đệ quy nếu búp bê A là búp bê lớn nhất của bộ. Em hãy cho biết dấu ? trong hàm **dem_Bupbe(búp bê A)** cần được thay bằng gì.

```
File  Edit  Format  Run  Options  Window  Help
def dem_Bupbe(búp bê A):
    if (không mở được búp bê A):      #Phần cơ sở
        return ?
    else:                             #Phần đệ quy
        return dem_Bupbe(con của búp bê A) + ?
```

Hình 3. Hàm mô tả cách đếm búp bê của bộ Matryoshka nếu búp bê A là búp bê lớn nhất của bộ

Định nghĩa đệ quy của một đối tượng là sự mô tả cách xây dựng đối tượng từ các phiên bản nhỏ hơn của chính đối tượng đó. Vì vậy, định nghĩa đệ quy của một đối tượng bao gồm hai phần sau:

- *Phần cơ sở:* phần định nghĩa đối tượng khi nó đã ở kích thước nhỏ nhất (không thể hoặc không cần thiết chia nhỏ hơn nữa).
- *Phần đệ quy:* phần chứa quy tắc để xây dựng đối tượng mới từ một đối tượng hoặc một số đối tượng cùng dạng có kích thước nhỏ hơn.

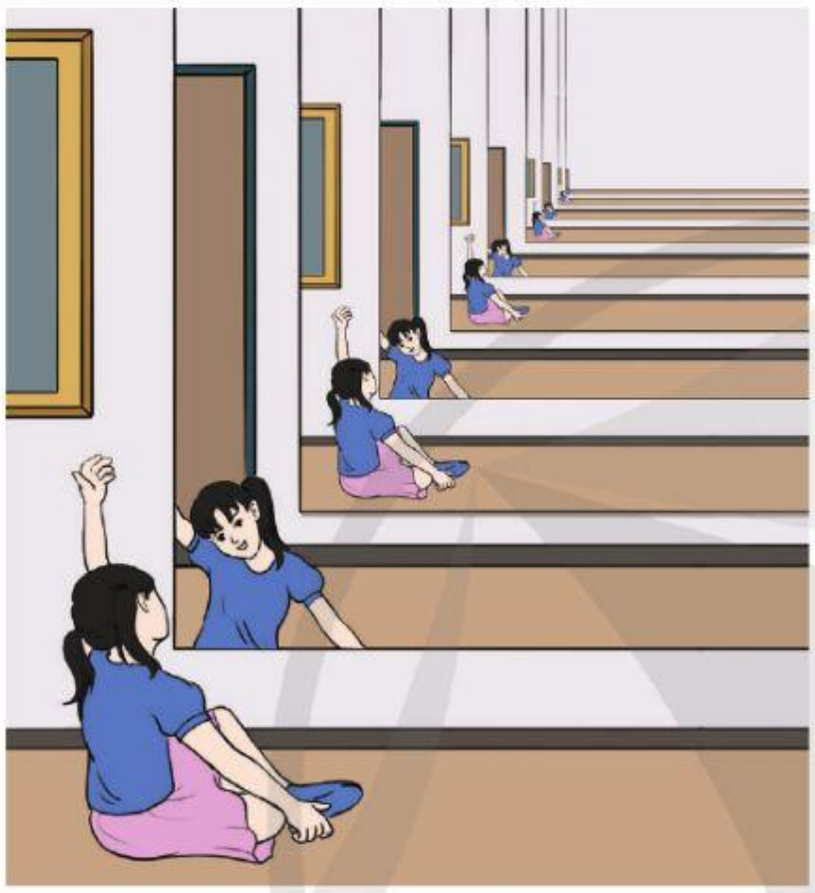
Đọc bản mới nhất trên hoc10.vn
Bản mẫu

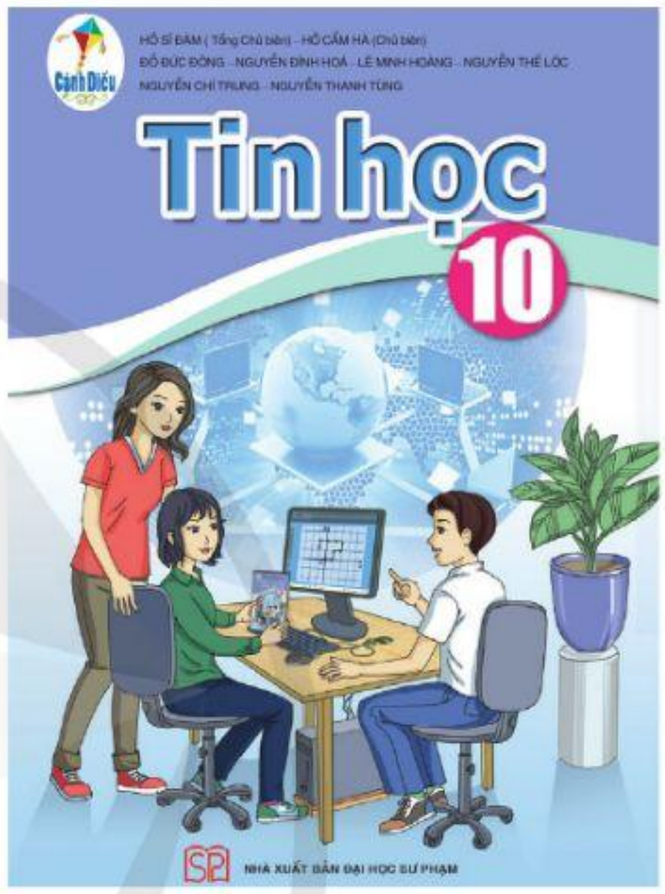
Câu 1. Xét tập S được định nghĩa đệ quy như sau:

- *Phần cơ sở:* 3 là phần tử của S.
- *Phần đệ quy:* Nếu x thuộc S và y thuộc S thì $x + y$ thuộc S (*chú ý:* x và y có thể có cùng giá trị).

Em hãy liệt kê 10 phần tử của tập S.

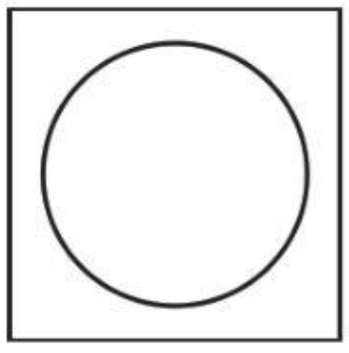
Câu 2. Công thức toán học, dãy số hay hình ảnh nào sau đây được xây dựng mang tính đệ quy? Tại sao?

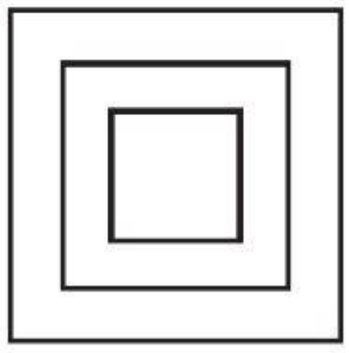
a)


b)


c)

d)

e)


g)


h)

i)

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

k) 1, 1, 2, –6, –24, 120,...

l) 4, 8, 12, 16, 20, 24, 28,...

m) $g(n) = \begin{cases} g(n-1) + g(n-2) + 2n \text{ nếu } n \geq 2 \\ 1 \text{ nếu } n = 0 \\ 2 \text{ nếu } n = 1 \end{cases}$

n) $f(n) = \begin{cases} 3^n + 2n - 1 \text{ nếu } n > 2 \\ 1 \text{ nếu } n = 1 \text{ hoặc } n = 2. \end{cases}$

Câu 3. Kí hiệu tập hợp tất cả các số nguyên dương lẻ là S. Em hãy:

a) Đưa ra định nghĩa đệ quy cho tập S.

b) So sánh cách mô tả tập S sử dụng định nghĩa đệ quy mà em xây dựng được ở câu a) với hai cách sau đây:

- Cách liệt kê các phần tử: $S = \{1, 3, 5, 7, 9, \ldots\}$.
- Cách sử dụng mệnh đề logic: $S = \{x \mid x \in \mathbb{N}^*, x$ không chia hết cho $2\}$.

Trong phòng họp có n người, mỗi người bắt tay lần lượt $n - 1$ người còn lại, giữa hai người bất kì chỉ bắt tay nhau đúng một lần. Em hãy:

a) Xác định số lượng cái bắt tay diễn ra trong phòng khi $n = 0, 1, 2, 3, 4$.

b) Đưa ra định nghĩa đệ quy cho hàm $h(n)$ tính số lượng cái bắt tay đã diễn ra trong phòng có n người.

Gợi ý: Để xây dựng phần đệ quy cho $h(n)$, em hãy xác định lời giải của bài toán khi có n người trong phòng từ lời giải của bài toán khi có $n - 1$ người trong phòng.

Trong những câu sau đây, câu nào đúng khi nói về đệ quy?

a) Ưu điểm của đệ quy là giúp cho mô tả đối tượng, sự việc trở nên ngắn gọn.

b) Khi đưa ra định nghĩa đệ quy của một đối tượng, không nhất thiết phải có phần cơ sở.

c) Trong một công thức đệ quy, phần đệ quy đưa ra quy tắc xây dựng đối tượng mới chỉ từ một đối tượng cùng dạng có kích thước nhỏ hơn.

d) Trong một công thức đệ quy, phần cơ sở là phần chứa quy tắc để xây dựng đối tượng mới từ một đối tượng cùng dạng có kích thước nhỏ hơn.

Tóm tắt bài học

Định nghĩa đệ quy của một đối tượng:

- Là sự mô tả cách xây dựng đối tượng từ các phiên bản nhỏ hơn của chính đối tượng đó.
- Bao gồm phần cơ sở và phần đệ quy.

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

Bài 2

THUẬT TOÁN ĐỆ QUY

Học xong bài này, em sẽ:

- Làm quen được với hàm đệ quy.
- Biết được các bước cần thực hiện khi giải bài toán bằng thuật toán đệ quy.
- Đọc hiểu được một vài chương trình tính toán đơn giản sử dụng thuật toán đệ quy.

Định nghĩa $a^n = \underbrace{a \times a \times \ldots \times a}_{n \text{ số } a}$. Em hãy đưa ra mô tả đệ quy cho hàm $F(n)$ để tính a^n.

1 Khái niệm hàm đệ quy

1

Hai chương trình cho trong *Hình 1* yêu cầu người sử dụng nhập hai giá trị nguyên a và n từ bàn phím ($n \geq 0$), rồi gọi hàm **power1(a,n)** và **power2(a,n)** tương ứng để in ra màn hình giá trị a^n. Em hãy đọc cả hai chương trình này và:

a) Cho biết kết quả thu được của hai chương trình khi giá trị của cặp (a, n) nhập vào lần lượt bằng (2, 4) và (3, 6).

b) Nhận xét về sự khác nhau giữa hai hàm **power1(a,n)** và **power2(a,n)**.

File Edit Format Run Options Window Help

```
1  def power1(a,n):
2    kq = 1
3    for i in range(1,n+1):
4      kq = kq*a
5    return kq
6
7  #Nhập giá trị nguyên a và n:
8  a = int(input('Nhập a = '))
9  n = int(input('Nhập n = '))
10 #Xuất ra kết quả của hàm:
11 print(a,'mũ',n,'=',power1(a,n))
```

File Edit Format Run Options Window Help

```
1  def power2(a,n):
2    if n == 0:    #Trường hợp cơ sở
3      return 1
4    else:
5      return a*power2(a,n-1) #Gọi đệ quy
6
7  #Nhập giá trị nguyên a và n:
8  a = int(input('Nhập a = '))
9  n = int(input('Nhập n = '))
10 #Xuất ra kết quả của hàm:
11 print(a,'mũ',n,'=',power2(a,n))
```

Hình 1. Hai chương trình tính a^n

Các ngôn ngữ lập trình bậc cao thường cho phép xây dựng các **hàm đệ quy**, nghĩa là trong thân của hàm có chứa những câu lệnh gọi đến chính nó.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

2 Thuật toán đệ quy

Việc phát triển thuật toán đệ quy là thuận tiện khi cần xử lí với các đối tượng được định nghĩa đệ quy. Khi cài đặt các thuật toán đệ quy, người ta thường xây dựng các hàm đệ quy. Thuật toán đệ quy thường có mô hình như ở *Hình 2*.

```
def deQuy(<input>):
    if (<kích thước của input là nhỏ nhất>): #Trường hợp cơ sở
        <Giải bài toán kích thước input nhỏ nhất để thu được lời_giải>
    else:
        deQuy(<input với kích thước nhỏ hơn>) #Gọi đệ quy
        <Kết hợp lời giải của các bài toán có kích thước input nhỏ hơn
        để thu được lời_giải>
    return <lời_giải>
```

Hình 2. Mô hình của thuật toán đệ quy

Khi phát triển thuật toán đệ quy cho một bài toán, ta cần xác định các trường hợp đơn giản nhất của bài toán mà ta có thể giải một cách trực tiếp. Các trường hợp này được gọi là các trường hợp cơ sở hoặc các trường hợp dừng của đệ quy. Khi giải các trường hợp khác, ta cần thực hiện các lời gọi đệ quy để giải quyết các bài toán cùng dạng với kích thước nhỏ hơn (gọi là các bài toán con). Lời gọi đệ quy này lại gọi đến lời đệ quy khác và đến một lúc nào đó các lời gọi đệ quy phải dẫn đến trường hợp cơ sở, lúc đó lời gọi đệ quy được kết thúc. Cuối cùng, kết hợp lời giải của các bài toán con sẽ thu được lời giải của bài toán ban đầu.

2

Em hãy:

a) Đọc chương trình ở *Hình 3* và cho biết dấu ? trong hàm `h(n)` cần được thay bằng gì để tính được số lượng cái bắt tay diễn ra trong phòng họp có *n* người.

b) Chạy chương trình để tính số cái bắt tay khi $n = 5$ và $n = 10$.

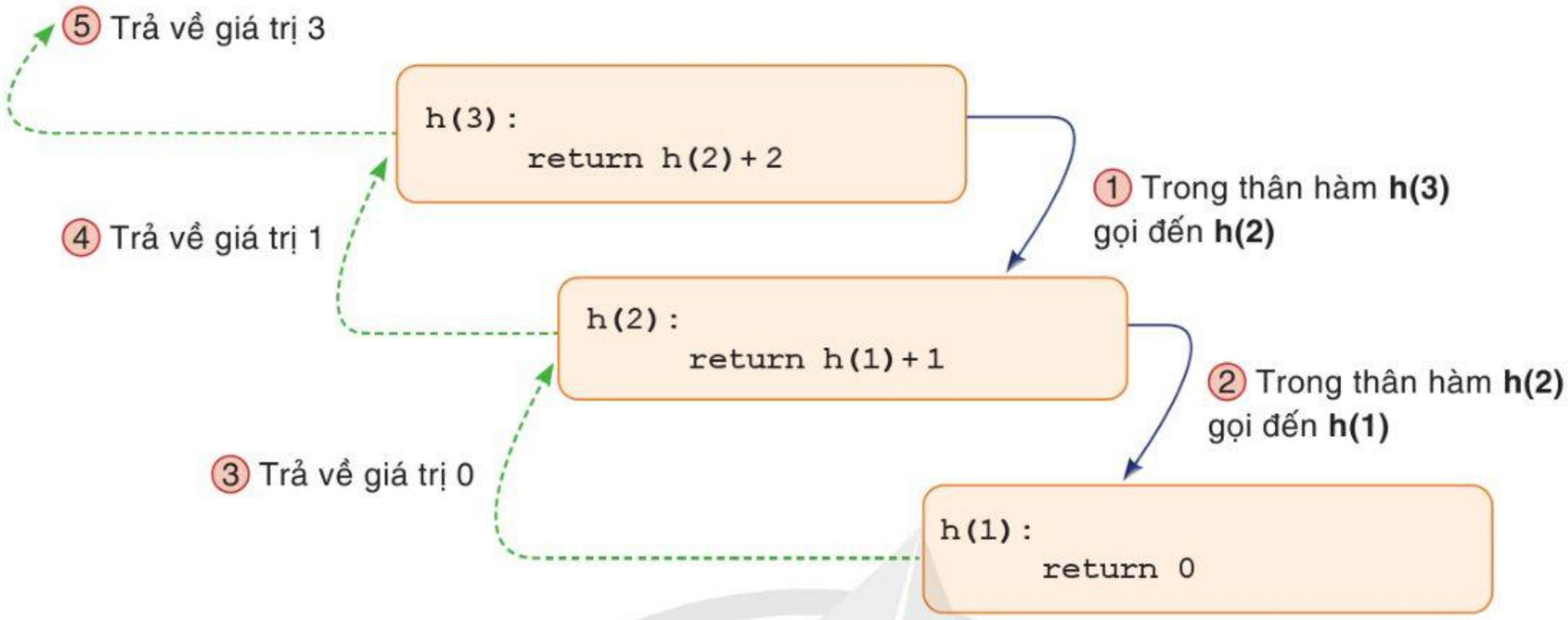
File Edit Format Run Options Window Help

```
1  def h(n):
2      if (n==1 or n==0):     #Trường hợp cơ sở
3          return ?
4      else:
5          return h(n-1) + ? #Gọi đệ quy
6
7  #Nhập giá trị n:
8  n = int(input('Nhập số lượng người trong phòng n = '))
9  #Xuất ra kết quả của hàm:
10 print('Tổng số cái bắt tay giữa',n,'người = ',h(n))
```

Hình 3. Chương trình đệ quy giải bài toán về tính số cái bắt tay

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Hàm **h(n)** ở *Hình 3* là một ví dụ về hàm đệ quy. *Hình 4* liệt kê lần lượt 5 bước chương trình sẽ thực hiện khi lời gọi đến **h(3)** được thực thi:



Hình 4. Sơ đồ phân tích trình tự hoạt động của chương trình khi thực thi ***h(3)***

Có thể thấy, bản chất của hàm đệ quy là trong thân hàm gọi một hoặc nhiều thực thi khác của hàm để tính toán kết quả của trường hợp kích thước nhỏ hơn, lấy kết quả đó kết hợp với xử lí tính toán trong phần thực thi hiện tại để đưa ra kết quả cuối cùng:

• Tại *Bước 1* và *Bước 2*, chương trình gọi đệ quy lần lượt đến **h(2)**, **h(1)**. Khi **h(1)** được gọi, tức là ta đã đạt đến trường hợp cơ sở trong định nghĩa đệ quy của **h(*n*)**, tương ứng với tham số $n = 1$. Câu lệnh kiểm tra điều kiện `if (n == 1 or n == 0)` trả về giá trị đúng, hàm **h(1)** trả về giá trị 0 tại *Bước 3*.

• Thoát khỏi **h(1)**, chương trình quay lại hàm trước đó là **h(2)**, thực hiện câu lệnh **h(1) + 1**, tức là 0 + 1, như vậy hàm **h(2)** kết thúc và trả về giá trị 1 tại *Bước 4*.

• Cuối cùng, sau khi thoát khỏi **h(2)**, chương trình quay lại hàm trước đó là **h(3)**, thực hiện câu lệnh **h(2) + 2**, tức là 1 + 2. Hàm **h(3)** trả về giá trị 3 tại *Bước 5*. Chương trình kết thúc.

Câu 1. Em hãy đọc chương trình ở *Hình 5* và cho biết kết quả nhận được khi chạy chương trình.

File Edit Format Run Options Window Help

```
1 def test(n):
2     if n<=0:    #Trường hợp cơ sở
3         return 0
4     if n%2==0:
5         return n + test(n-2) #Gọi đệ quy
6     else:
7         return test(n-1) #Gọi đệ quy
8 print('Kết quả =', test(6))
```

Hình 5. Chương trình sử dụng ngôn ngữ lập trình Python

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Câu 2. Dãy số Fibonacci được định nghĩa đệ quy như sau:

• *Phần cơ sở:* $F(n) = 0$ nếu $n = 0$, $F(n) = 1$ nếu $n = 1$.

• *Phần đệ quy:* $F(n) = F(n-1) + F(n-2)$ nếu $n \geq 2$.

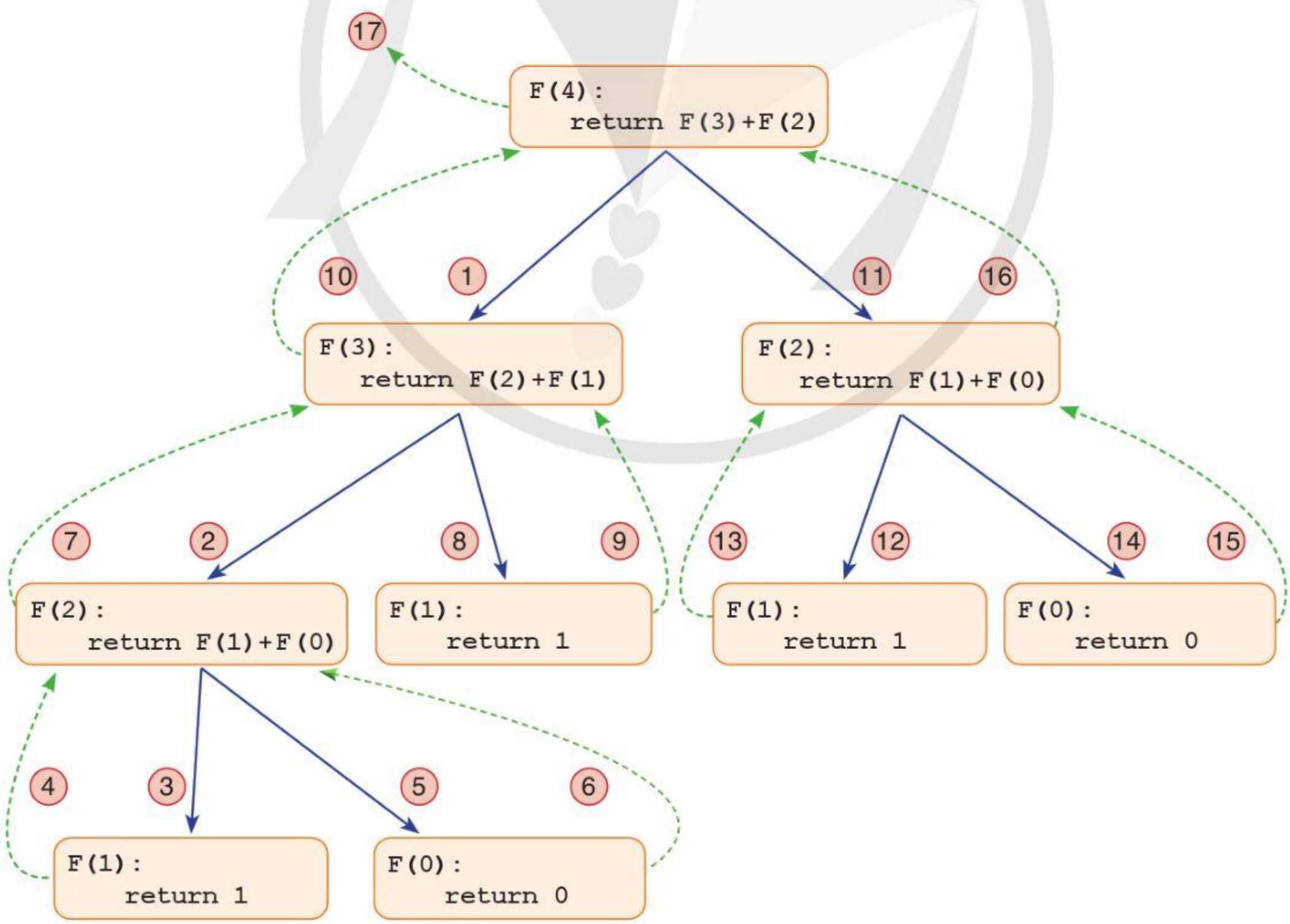
Hàm đệ quy **F(n)** cho trong *Hình 6* sử dụng định nghĩa đệ quy ở trên để tính và trả về giá trị của $F(n)$.

a) Em hãy cho biết các dấu ? trong hàm đệ quy **F(n)** cần được thay bằng gì.

b) *Hình 7* liệt kê lần lượt 17 bước chương trình sẽ thực hiện khi lời gọi đến **F(4)** được thực thi. Em hãy đưa ra giải thích bằng lời ý nghĩa của 17 bước đã cho.

```
File  Edit  Format  Run  Options  Window  Help
1  def F(n):
2      if n in {0, 1}: #Trường hợp cơ sở
3          return ?
4      else:
5          return F(n-1) ? #Gọi đệ quy
```

Hình 6. Hàm đệ quy **F(n)**



Hình 7. Sơ đồ phân tích trình tự hoạt động của chương trình khi thực thi **F(4)**

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

Mẹ An là cô giáo dạy Toán, muốn nhờ An in lời một bài hát dạy đếm số cho các em học sinh có định dạng như sau:

Có 20 chiếc kẹo trong hộp các bạn ơi, ta lấy 1 chiếc ra ăn các bạn nhé,
Vậy trong hộp còn 19 chiếc thôi.
Có 19 chiếc kẹo trong hộp các bạn ơi, ta lấy 1 chiếc ra ăn các bạn nhé,
Vậy trong hộp còn 18 chiếc thôi.

...

Có 2 chiếc kẹo trong hộp các bạn ơi, ta lấy 1 chiếc ra ăn các bạn nhé,
Vậy trong hộp còn 1 chiếc thôi.
Còn duy nhất 1 chiếc kẹo trong hộp các bạn ơi, ta lấy nốt chiếc cuối cùng ra ăn các bạn nhé,
Không còn chiếc kẹo nào trong hộp, ta cùng nhau đi mua thêm kẹo thôi nào.

Em hãy giúp An viết hàm đệ quy `in_loi_bai_hat(n)` với `n` là số kẹo để có thể in ra lời bài hát như trên nếu gọi hàm `in_loi_bai_hat(20)`.

Trong những câu sau đây, câu nào đúng khi nói về hàm đệ quy?

a) Trong hàm đệ quy chỉ được phép chứa duy nhất một lệnh gọi đến chính nó.
b) Trong hàm đệ quy, trường hợp cơ sở không được phép tiếp tục gọi đệ quy.
c) Hàm đệ quy là hàm mà thân hàm có chứa những lệnh gọi đến chính nó.
d) Hàm đệ quy được sử dụng để cài đặt thuật toán đệ quy.
e) Trong hàm đệ quy có thể có nhiều hơn một trường hợp cơ sở.
g) Hàm đệ quy không bao giờ dừng nếu không có trường hợp cơ sở.

Tóm tắt bài học

- Đặc điểm của hàm đệ quy:
  - Trong hàm có một hoặc nhiều lệnh gọi đến chính nó.
  - Mỗi lần gọi đệ quy thì kích thước của bài toán được thu nhỏ hơn so với lần gọi trước. Khi đạt được trường hợp cơ sở thì chương trình không cần gọi đệ quy.
- Thuật toán đệ quy được cài đặt dưới dạng hàm đệ quy, để xử lí với các đối tượng được định nghĩa đệ quy.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

BÀI TÌM HIỂU THÊM

ĐỆ QUY CÓ NHỚ

Trong *Hình 7*, ta thấy để tính giá trị **F(4)**, chương trình phải gọi tính hàm **F(2)** hai lần ở bước thứ 2 và 11; gọi tính hàm **F(1)** ba lần ở bước thứ 3, 8 và 12; gọi hàm **F(0)** hai lần ở bước thứ 5 và 14. Điều đó cho thấy, thuật toán đệ quy để tính số Fibonacci là chưa hiệu quả vì có những bài toán con bị giải đi giải lại nhiều lần.

Để tăng hiệu quả của thuật toán đệ quy mà vẫn có thể giữ nguyên cấu trúc đệ quy của thuật toán, ta có thể sử dụng kĩ thuật **đệ quy có nhớ**. Ý tưởng của kĩ thuật này là: Ta sẽ dùng biến ghi nhớ lại thông tin về lời giải của các bài toán con ngay sau lần đầu tiên nó được giải. Điều đó cho phép rút ngắn thời gian tính của thuật toán, bởi vì, mỗi khi cần đến có thể tra cứu mà không phải giải lại những bài toán con đã được giải trước đó. Tuy nhiên, nhược điểm lớn nhất của cách làm này là đòi hỏi về bộ nhớ.

File Edit Format Run Options Window Help

```
1  memory = {0:0,1:1}
2
3  def Fmem(n):
4      if n not in memory:
5          memory[n] = Fmem(n-1) + Fmem(n-2)
6      return memory[n]
7
8  n = int(input('Nhập số nguyên dương n = '))
9  print('Kết quả F(',n,') = ',Fmem(n))
```

*Hình 8. Hàm đệ quy có nhớ **Fmem(n)** tính và trả về giá trị của hàm **F(n)***

Chương trình ở *Hình 8* sử dụng kĩ thuật đệ quy có nhớ để cài đặt hàm đệ quy **Fmem(n)** tính số Fibonacci thứ n với $n = 0, 1, 2,...$ Ta đưa vào biến **memory[n]** để ghi nhận giá trị số Fibonacci thứ n. Nghĩa là, khi tính được số Fibonacci thứ n, giá trị này sẽ được ghi nhận vào **memory[n]**. Như vậy, nếu **memory[n]** đã có giá trị thì điều đó có nghĩa là số Fibonacci thứ n đã được tính, ta chỉ cần trả về giá trị **memory[n]** cho hàm **Fmem(n)** ở những lần gọi hàm tiếp theo mà không cần gọi đệ quy tới **Fmem(n-1)** và **Fmem(n-2)**.

Em hãy chạy **Fmem(n)** ở trên và hàm **F(n)** ở *Hình 6* với $n = 30$, $n = 40$. So sánh kết quả và thời gian chạy của **Fmem(30)** và **F(30)**, **Fmem(40)** và **F(40)**.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bài 3

THỰC HÀNH THIẾT KẾ THUẬT TOÁN ĐỆ QUY

Học xong bài này, em sẽ:

- Viết và thực hiện được một vài chương trình cài đặt thuật toán đệ quy.
- Nhận biết được lỗi lặp vô hạn khi cài đặt thuật toán đệ quy.

Bài toán 1. Tìm ước số chung lớn nhất

a) Để tìm ước số chung lớn nhất của hai số tự nhiên x và y ($x \neq 0$, $y \neq 0$), ta sử dụng công thức sau:

$$\text{UCLN}(x, y) = \begin{cases} y \text{ nếu } r = 0; \\ \text{UCLN}(y, r) \text{ nếu } r \neq 0; \end{cases}$$ với r là số dư trong phép chia x cho y.

Em hãy chạy chương trình ở *Hình 1* với một số bộ dữ liệu đầu vào (x, y) khác nhau để kiểm thử chương trình.

```
File  Edit  Format  Run  Options  Window  Help
def UCLN(x,y):
    r = x%y
    if (r==0):        #Trường hợp cơ sở
        return y
    else:
        return UCLN(y,r) #Gọi đệ quy

x = int(input('Nhập số tự nhiên khác 0 x = '))
y = int(input('Nhập số tự nhiên khác 0 y = '))
print('UCLN = ',UCLN(x,y))
```

Hình 1. Chương trình tìm ước số chung lớn nhất của hai số tự nhiên

Ví dụ:

DỮ LIỆU VÀO	KẾT QUẢ RA
14	UCLN = 7
21	

b) Viết hàm đệ quy `UCLN1(x,y)` tìm ước số chung lớn nhất của hai số tự nhiên x và y không đồng thời bằng 0, sử dụng công thức sau:

Đọc bản mới nhất trên hoc10.vn Bản mẫu

$$\text{UCLN}(x, y) = \begin{cases} x \text{ nếu } y = 0; \\ \text{UCLN}(y, r) \text{ nếu } y \neq 0; \text{ với } r \text{ là số dư trong phép chia } x \text{ cho } y. \end{cases}$$

Sau đó, viết chương trình gọi hàm **UCLN1(x,y)** để tìm ước số chung lớn nhất của hai số x và y được nhập vào từ bàn phím. Chạy chương trình với các bộ dữ liệu đầu vào em đã sử dụng ở câu a) và so sánh kết quả thu được.

c) Em hãy chạy chương trình trong câu a) và b) với hai bộ dữ liệu $x = 5$, $y = 0$ và $x = 0$, $y = 5$. Sau đó, nhận xét kết quả thu được.

Bài toán 2. Nhận biết lỗi lặp vô hạn khi cài đặt đệ quy

Em hãy thực hiện các yêu cầu sau:

a) Đọc hàm đệ quy **product(n)** ở *Hình 2* và cho biết kết quả thu được khi thực thi lời gọi hàm **product(8)**.

b) Chạy chương trình ở *Hình 2* và cho biết Python thông báo lỗi như thế nào.

c) Sửa chương trình ở *Hình 2* như sau: Thay câu lệnh cuối cùng **print('KQ = ', product(8))** thành **print('KQ = ', product(9))** và cho biết kết quả khi chạy chương trình.

d) Theo em, tại sao Python lại thông báo lỗi khi chương trình thực thi **product(8)**, nhưng lại không báo lỗi khi chương trình thực thi **product(9)**?

```
File Edit Format Run Options Window Help
1 def product(n):
2     if (n==1):
3         return 1
4     else:
5         return n * product(n-2)
6
7 print('KQ = ',product(8))
```

Hình 2. Chương trình đệ quy có lỗi

Gợi ý: Kiểm tra xem trong quá trình thực hiện **product(8)** có lần gọi đệ quy nào đến được trường hợp cơ sở **if(n==1)** hay không.

Trong quá trình thực thi một hàm đệ quy mà bước cơ sở không bao giờ được thực hiện thì hàm đệ quy sẽ bị lặp vô hạn, không bao giờ dừng.

Bài toán 3. Xác định ý nghĩa của hàm đệ quy cho trước

Em hãy chạy chương trình ở *Hình 3* và cho biết hàm đệ quy **mystery(n)** với đầu vào n là số nguyên được dùng để giải quyết bài toán nào.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

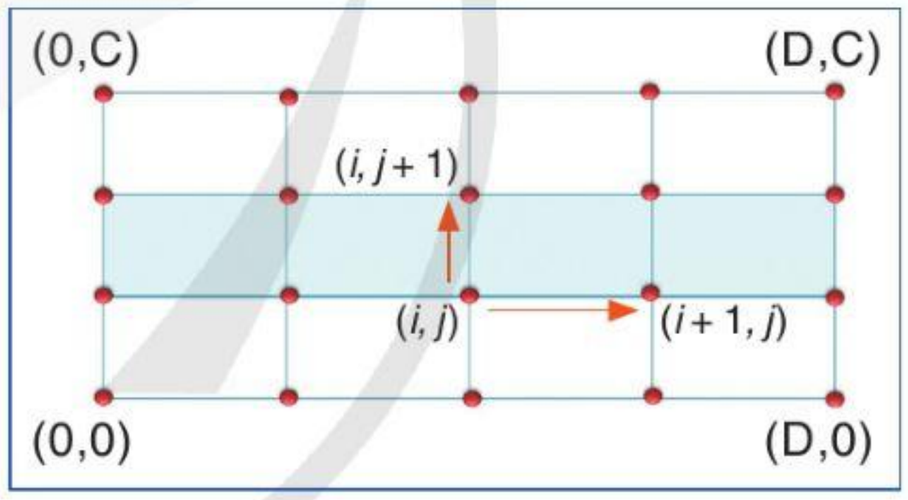
```
File Edit Format Run Options Window Help
def mystery(n):
    t = abs(n//10)
    if t==0 :
        return 1
    else:
        return 1 + mystery(t)

print(mystery(-1567))
print(mystery(1567))
print(mystery(123))
```

Hình 3. Chương trình có sử dụng hàm đệ quy

a) Tìm hiểu bài toán:

Một cánh đồng được chia thành các thửa ruộng hình chữ nhật như *Hình 4*. Bờ ruộng được thể hiện bởi các đường thẳng màu xanh. Các nút tròn đỏ trên hình thể hiện các nút giao lộ giữa các thửa ruộng, được đánh theo toạ độ từ $(0, 0)$ đến (D, C). Với mỗi bước nhảy bật xa của mình, An chỉ có thể đi từ nút giao lộ này sang nút giao lộ khác trên bờ ruộng theo hướng hoặc lên trên hoặc sang phải.

Cụ thể là, khi đang đứng ở nút giao lộ có toạ độ (i, j) bất kì với $0 \leq i \leq D$, $0 \leq j \leq C$, với một bước nhảy, An chỉ có thể nhảy tới nút $(i, j + 1)$ hoặc nút $(i + 1, j)$. Em hãy xác định giúp An có tất cả bao nhiêu cách để có thể đi từ nút xuất phát $(i, j) \neq (D, C)$ đến nút (D, C) bằng các bước nhảy của mình.



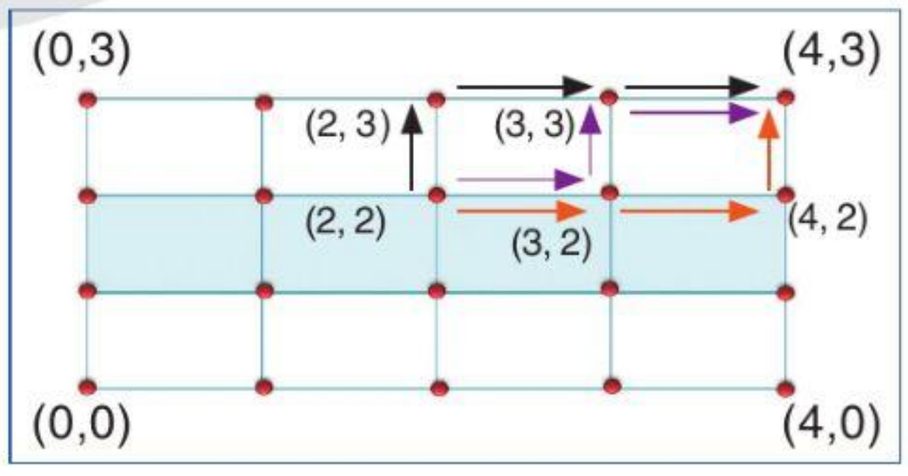
Hình 4. Cánh đồng với $D = 4$, $C = 3$

Ví dụ: Cho cánh đồng với $D = 4$, $C = 3$ như ở *Hình 5*. Giả sử An đang đứng ở nút (2, 2), thì để đến được nút (4, 3) An có thể đi theo một trong ba cách sau:

1) $(2, 2) \rightarrow (2, 3) \rightarrow (3, 3) \rightarrow (4, 3)$

2) $(2, 2) \rightarrow (3, 2) \rightarrow (3, 3) \rightarrow (4, 3)$

3) $(2, 2) \rightarrow (3, 2) \rightarrow (4, 2) \rightarrow (4, 3)$



Hình 5. Minh hoạ các cách đi trên cánh đồng $D = 4$, $C = 3$

b) Em hãy đọc hiểu và chạy chương trình ở *Hình 6* và cho biết chương trình này có giải quyết được bài toán trên hay không.

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

```
File Edit Format Run Options Window Help
def dem(i,j,D,C):
    if (i==D or j==C):#Trường hợp cơ sở
        return 1
    else:
        return dem(i+1,j,D,C)+ dem(i,j+1,D,C) #Gọi đệ quy

D = int(input('Nhập số nguyên dương D = '))
C = int(input('Nhập số nguyên dương C = '))
i = int(input('Nhập số nguyên không âm i = '))
j = int(input('Nhập số nguyên không âm j = '))
print('Số cách đi = ',dem(i,j,D,C))
```

```
Nhập số nguyên dương D = 4
Nhập số nguyên dương C = 3
Nhập số nguyên không âm i = 2
Nhập số nguyên không âm j = 2
Số cách đi = 3
```

Hình 6. Chương trình giải bài toán đếm số đường đi trên cánh đồng và màn hình kết quả chạy một bộ dữ liệu thử nghiệm

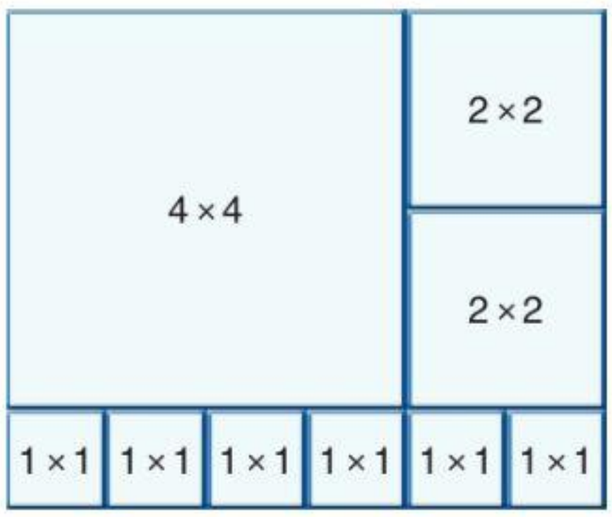
BÀI TÌM HIỂU THÊM

TÌM CÁCH LÁT GẠCH TỐI ƯU

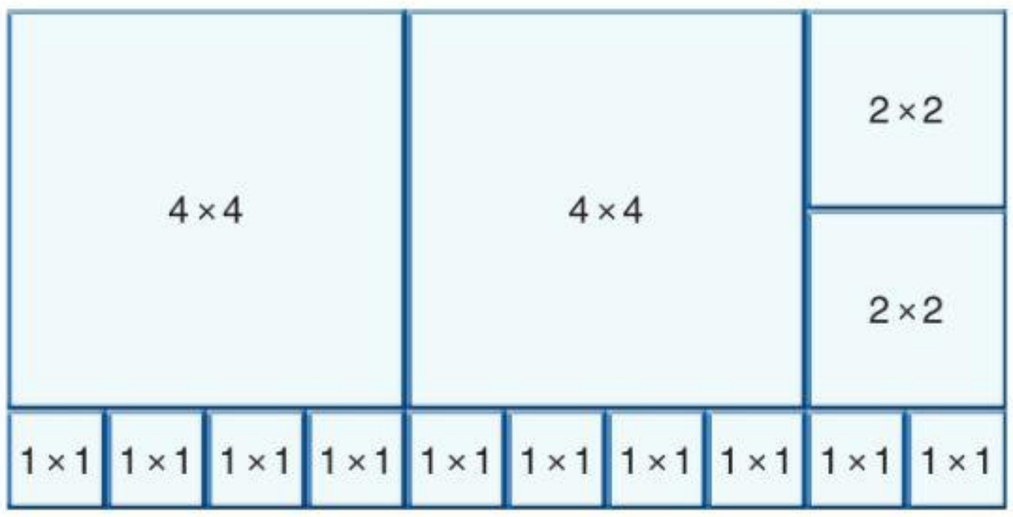
Gia đình An muốn lát một mảnh đất hình chữ nhật kích thước $N \times M$ (dm^2) bởi các viên gạch hình vuông. Gia đình An yêu cầu các viên gạch hình vuông này phải có kích thước dạng $2^i \times 2^i$, $i = 0, 1, 2, 3, \ldots$ Em hãy giúp An xác định số lượng gạch tối thiểu cần dùng để có thể lát được mảnh đất của gia đình bằng cách viết hàm đệ quy **`soGachMin(D,R)`** trả về số lượng gạch tối thiểu cần dùng nếu mảnh đất có chiều dài là D (dm) và chiều rộng là R (dm).

Ví dụ 1: Nếu $N = 5, M = 6$ thì ta cần tối thiểu là 9 viên gạch gồm 6 viên kích thước 1×1, 2 viên kích thước 2×2, 1 viên kích thước 4×4, với cách lát như ở *Hình 7a*.

Ví dụ 2: Nếu $N = 5, M = 10$ thì ta cần tối thiểu là 14 viên gạch gồm 10 viên kích thước 1×1, 2 viên kích thước 2×2, 2 viên kích thước 4×4, với cách lát như ở *Hình 7b*.



Hình 7a. Cách lát cho mảnh đất 5 × 6



Hình 7b. Cách lát cho mảnh đất 5 × 10

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

Bài 4

THỰC HÀNH TỔNG HỢP THIẾT KẾ THUẬT TOÁN ĐỆ QUY

Học xong bài này, em sẽ:

- Hiểu được các bước giải bài toán Tháp Hà Nội sử dụng thuật toán đệ quy.
- Viết và thực hiện được chương trình cài đặt thuật toán đệ quy giải bài toán Tháp Hà Nội.

Bài toán *Tháp Hà Nội*

Bài toán Tháp Hà Nội được trình bày dưới dạng trò chơi như sau: Có ba cọc A, B, C. Trên cọc A có một chồng đĩa gồm *n* cái đĩa, đường kính giảm dần từ dưới lên trên. Cần phải chuyển chồng đĩa từ cọc A sang cọc C theo quy tắc:

1) Mỗi lần chỉ chuyển một đĩa ở trên cùng của một cọc.

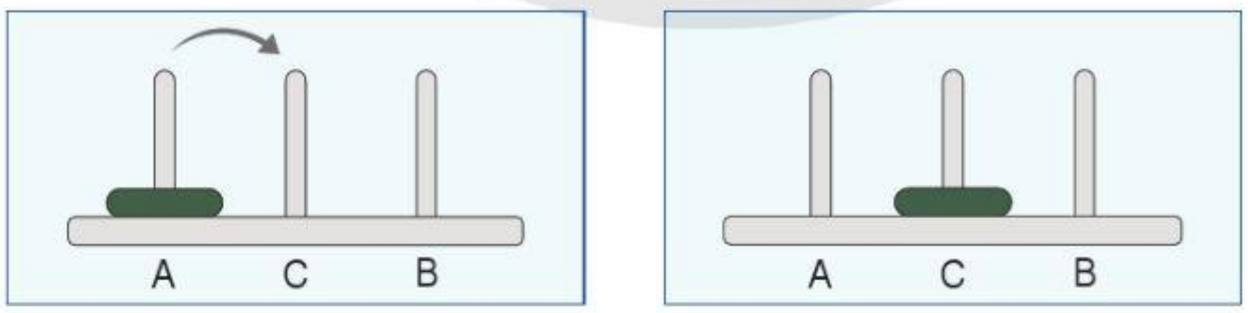
2) Chỉ được xếp đĩa có đường kính nhỏ hơn lên trên đĩa có đường kính lớn hơn. Trong quá trình chuyển được phép dùng cọc B làm cọc trung gian.

Bài toán đặt ra là: Tìm cách chơi và đưa ra từng bước di chuyển đĩa thoả mãn yêu cầu.

Các *Hình 1, 2, 3* lần lượt minh hoạ các bước di chuyển đĩa cần thực hiện khi số đĩa $n = 1, 2, 3$.

Với $n = 1$: Ta chỉ cần duy nhất một bước di chuyển đĩa từ cọc A sang cọc C.

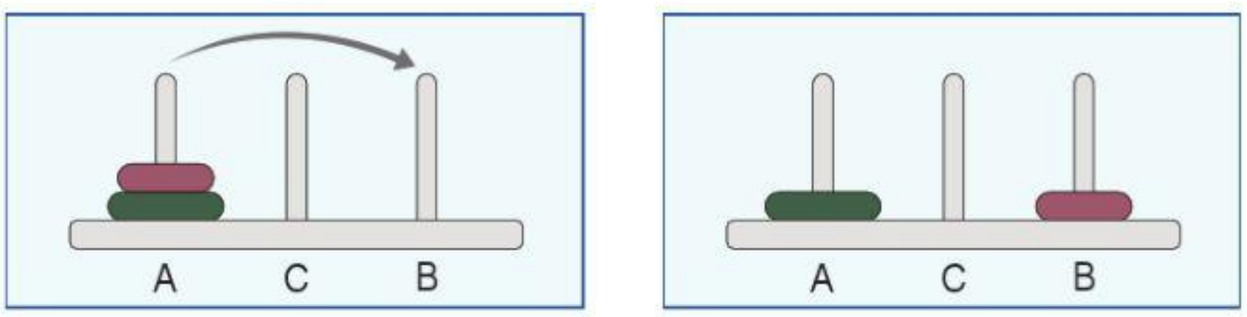
Bước 1



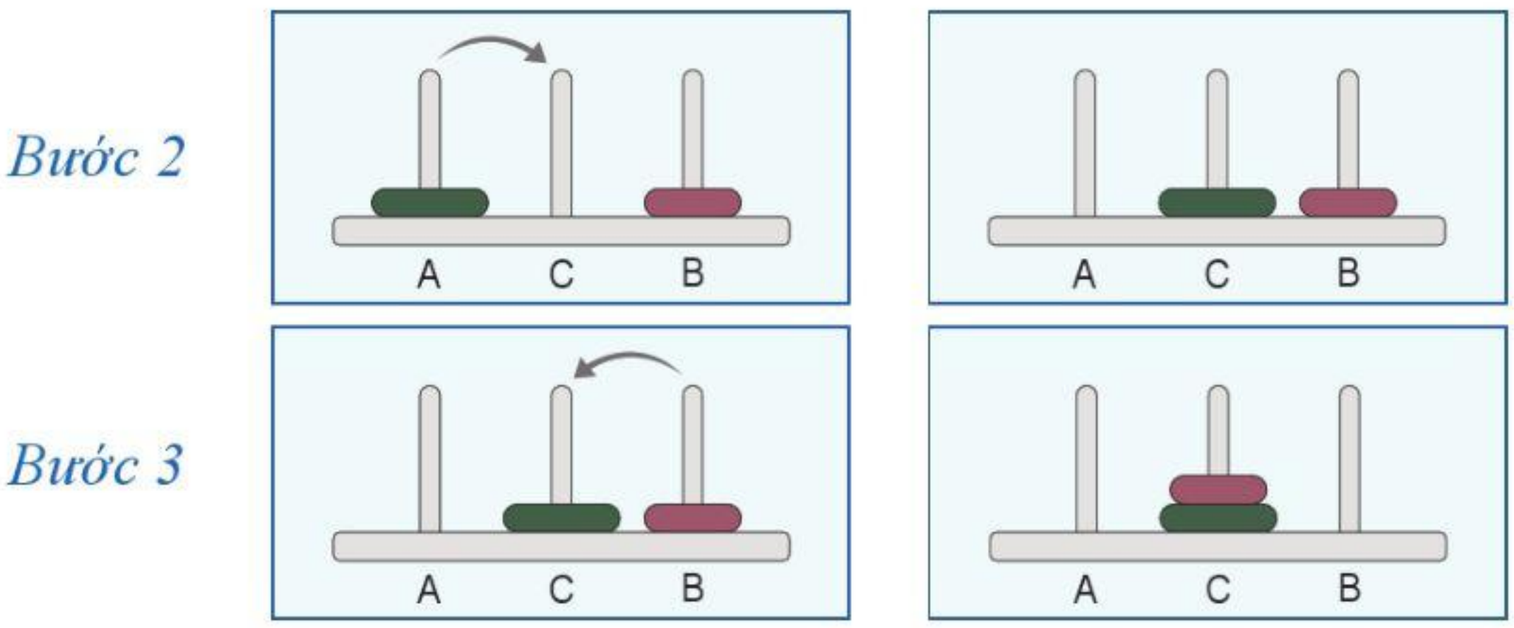
Hình 1. Các bước di chuyển đĩa khi n = 1

Với $n = 2$: Ta cần 3 bước di chuyển đĩa.

Bước 1

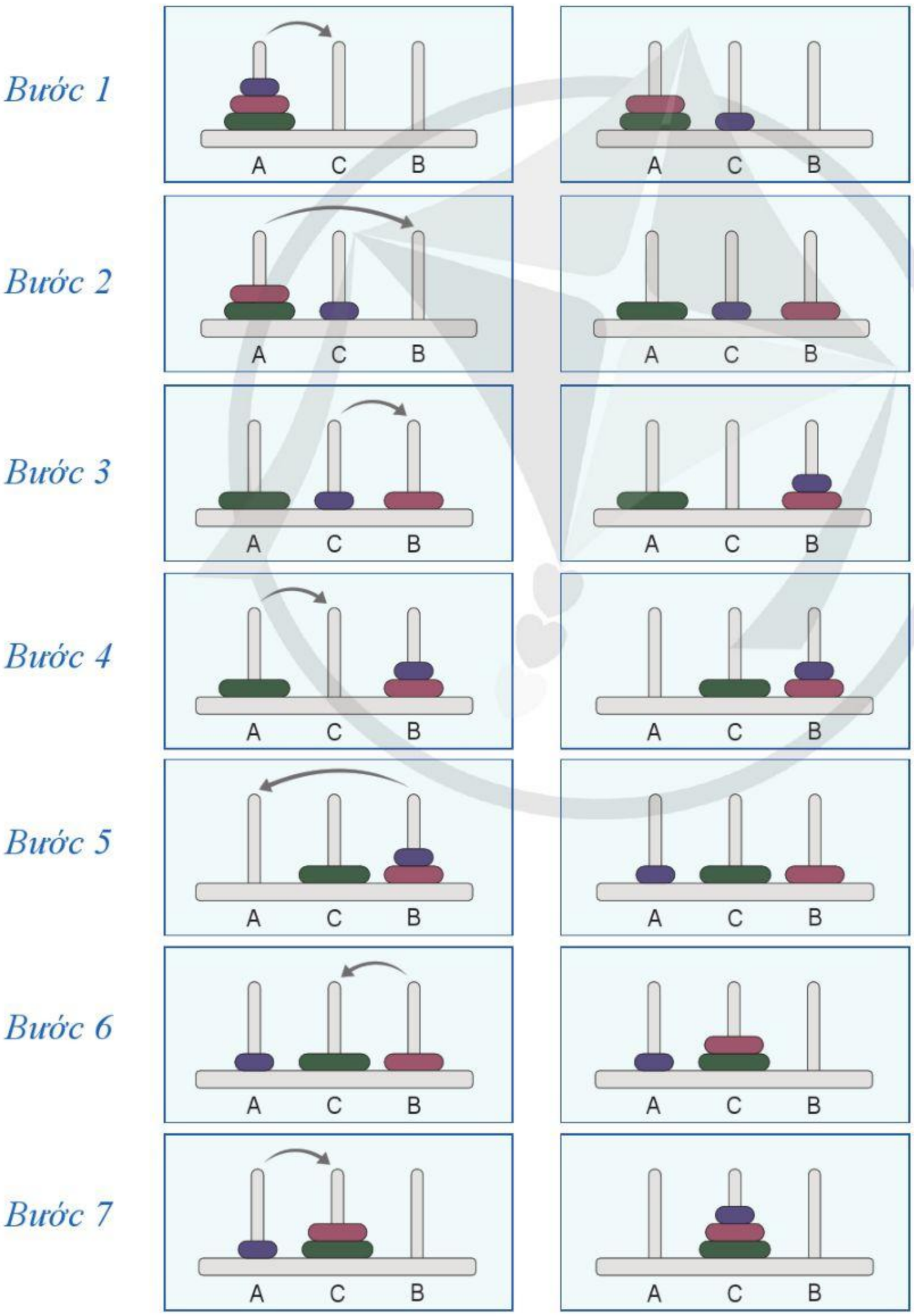


Đọc bản mới nhất trên hoc10.vn
Bản mẫu



Hình 2. Các bước di chuyển đĩa khi n = 2

Với $n = 3$: Ta cần 7 bước di chuyển đĩa.



Hình 3. Các bước di chuyển đĩa khi n = 3

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

a) Trong quá trình di chuyển đĩa gồm 7 bước với $n = 3$, nhận thấy bài toán Tháp Hà Nội cho trường hợp $n = 2$ được giải hai lần: lần giải đầu tiên bởi ba bước *1*, *2*, *3* và lần giải thứ hai bởi ba bước *5*, *6*, *7*. Sau ba bước *1*, *2*, *3* hai đĩa trên cùng của cọc A được chuyển sang cọc B. Do đó, ở lần giải đầu tiên này, cọc A được gọi là cọc xuất phát, cọc B được gọi là cọc đích. Em hãy nêu tên cọc xuất phát và cọc đích ở lần giải thứ hai tương ứng với ba bước *5*, *6*, *7*.

b) Với $n = 4$, giả sử đã chuyển được ba đĩa trên cùng của cọc A sang cọc B. Em hãy thực hiện tiếp các bước để cả 4 đĩa đều ở cọc C và cho biết khi giải bài toán Tháp Hà Nội với $n = 4$ ta cần giải bao nhiêu lần bài toán này với $n = 3$.

c) Xây dựng thuật toán đệ quy giải quyết bài toán Tháp Hà Nội với n đĩa và cài đặt thuật toán đề xuất bằng một hàm đệ quy.

d) Viết chương trình yêu cầu người dùng nhập vào số lượng đĩa n và gọi hàm đệ quy đã xây dựng được, để hướng dẫn người chơi các bước di chuyển đĩa. Sau đó, em hãy chạy thử chương trình với các giá trị n lần lượt là 3, 4, 5 để kiểm thử chương trình.

Xét bài toán Tháp Hà Nội trong trường hợp cọc A có một chồng đĩa gồm $2n$ cái đĩa với n kích thước khác nhau (mỗi kích thước có hai cái đĩa), đường kính giảm dần từ dưới lên trên. Em hãy thực hiện các yêu cầu sau:

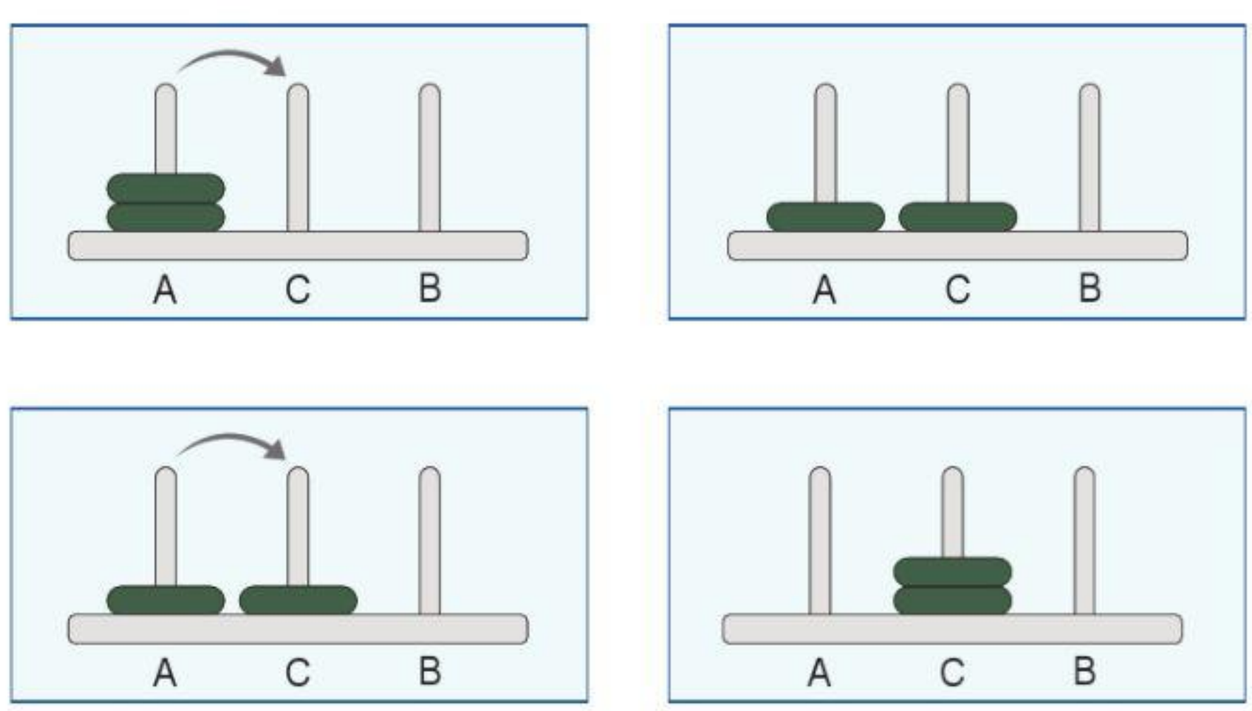
a) *Hình 4* và *5* minh hoạ cách di chuyển đĩa với $n = 1$ và $n = 2$ tương ứng. Bài toán với $n = 2$ có 6 bước di chuyển đĩa, em hãy cho biết trong đó có bao nhiêu lần giải bài toán với $n = 1$, nêu tên cọc xuất phát và cọc đích ở từng lần giải đó.

b) Khi giải bài toán với $n = 3$ thì phải giải bài toán với n nhỏ hơn nào, nêu tên cọc xuất phát và cọc đích ở từng lần giải đó.

c) Viết hàm đệ quy giải quyết bài toán. Kết quả là hiển thị các bước di chuyển đĩa. Sau đó, chạy hàm này với n lần lượt là 3, 4, 5 và kiểm tra kết quả thu được.

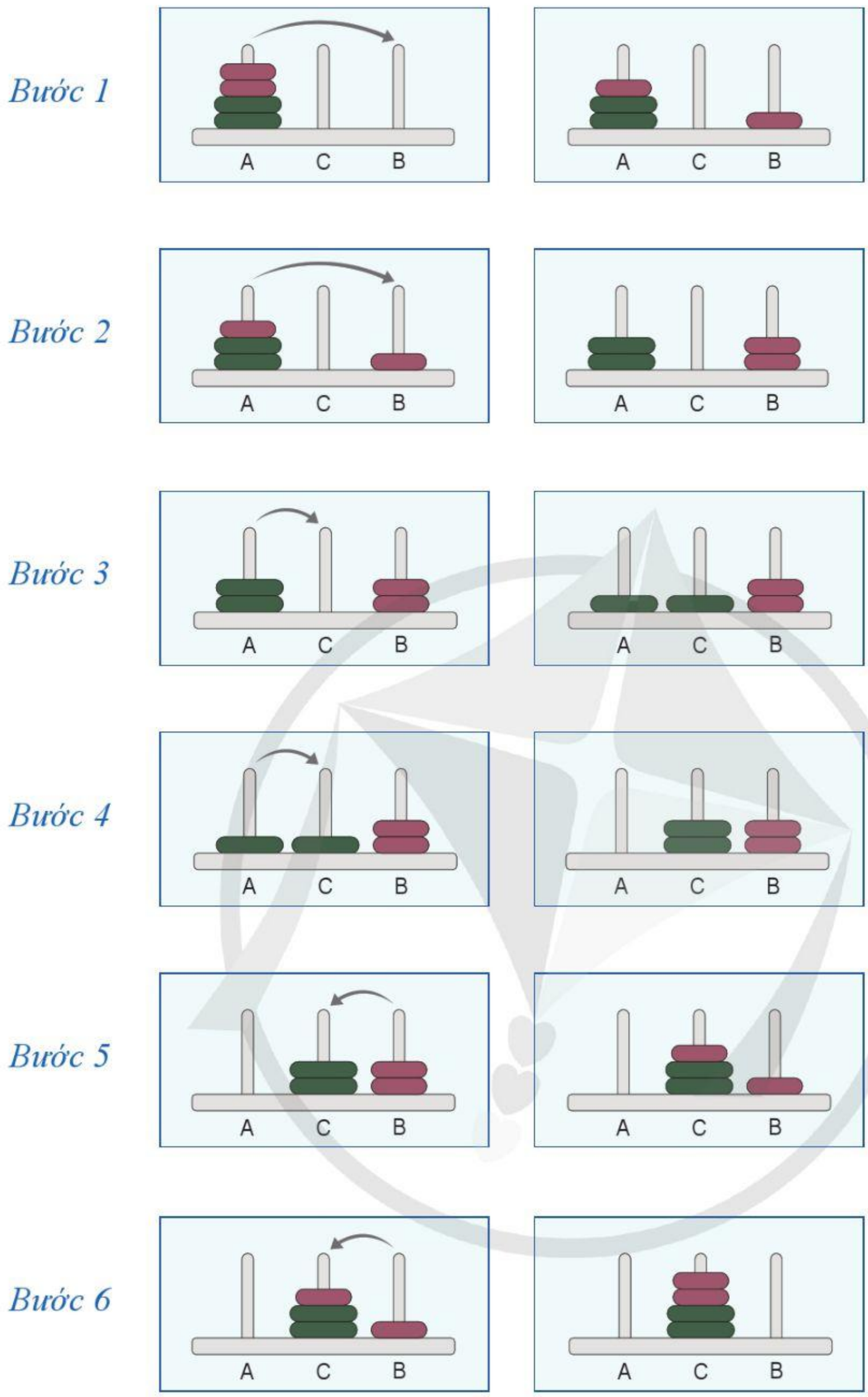
Bước 1

Bước 2



Hình 4. Các bước di chuyển đĩa khi n = 1

Đọc bản mới nhất trên hoc10.vn
Bản mẫu



Hình 5. Các bước di chuyển đĩa khi n = 2

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

CHUYÊN ĐỀ 2

THỰC HÀNH THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT CHIA ĐỂ TRỊ

Bài 1

Ý TƯỞNG CHIA ĐỂ TRỊ

Học xong bài này, em sẽ:

- Nêu được ý tưởng kĩ thuật chia để trị và thu hẹp phạm vi tìm kiếm.
- Hiểu được một số bài toán sử dụng kĩ thuật chia để trị.
- Cài đặt được thuật toán tìm kiếm nhị phân bằng vòng lặp.

Trong sách *Tin học 7*, em đã học thuật toán tìm kiếm nhị phân. Thuật toán này là một kĩ thuật thu hẹp phạm vi tìm kiếm trong phương pháp chia để trị. Em hãy quan sát dãy 9 số được sắp xếp tăng dần sau:

4 7 8 20 21 22 36 77 81

Số 21 ở vị trí chính giữa của dãy, các số bên trái của số 21 luôn nhỏ hơn 21 và các số bên phải của số 21 luôn lớn hơn 21. Do đó, nếu muốn tìm một số x nhỏ hơn 21 thì chỉ cần thu hẹp phạm vi tìm kiếm vào một nửa của dãy, theo em đó là nửa dãy bên trái hay nửa dãy bên phải của số 21?

1 Ý tưởng kĩ thuật chia để trị

1

Hai mô tả sau đây chỉ ra phương pháp hiệu quả giải quyết bài toán bổ và đếm số hạt dưa bằng ý tưởng kĩ thuật chia để trị. Em hãy tìm hiểu bài toán sau đây và rút ra ý tưởng chủ đạo của kĩ thuật chia để trị để giải quyết bài toán.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bài toán *Bổ dưa và đếm số hạt dưa*

Hôm nay là sinh nhật của Thanh An, có tất cả 8 bạn tham dự. Thanh An có một quả dưa hấu muốn chia thành 8 miếng dưa đều nhau và muốn biết tổng số hạt của quả dưa này. Để các miếng dưa có thể đều nhau, Thanh An nghĩ ra cách bổ đôi theo từng lượt (*Hình 1*).

Lượt 1: Bổ đôi quả dưa thành hai miếng (A/B).

Lượt 2: Bổ đôi mỗi miếng (A) và (B) thành hai miếng (A.1/A.2) và (B.1/B.2).

Lượt 3: Bổ đôi mỗi miếng (A.1), (A.2), (B.1), (B.2) thành hai miếng (A.1.1/A.1.2), (A.2.1/A.2.2), (B.1.1/B.1.2), (B.2.1/B.2.2).

Nhận xét: Việc bổ đôi miếng dưa hấu ở mỗi bước giúp ước lượng các miếng đều nhau dễ dàng hơn nhiều so với việc cắt miếng dưa thành những miếng nhỏ. Sau mỗi lượt bổ, số lượng miếng dưa tăng lên gấp đôi. Cách làm này thể hiện hai đặc điểm:

1. Chia nhỏ miếng dưa thành hai miếng nhỏ.

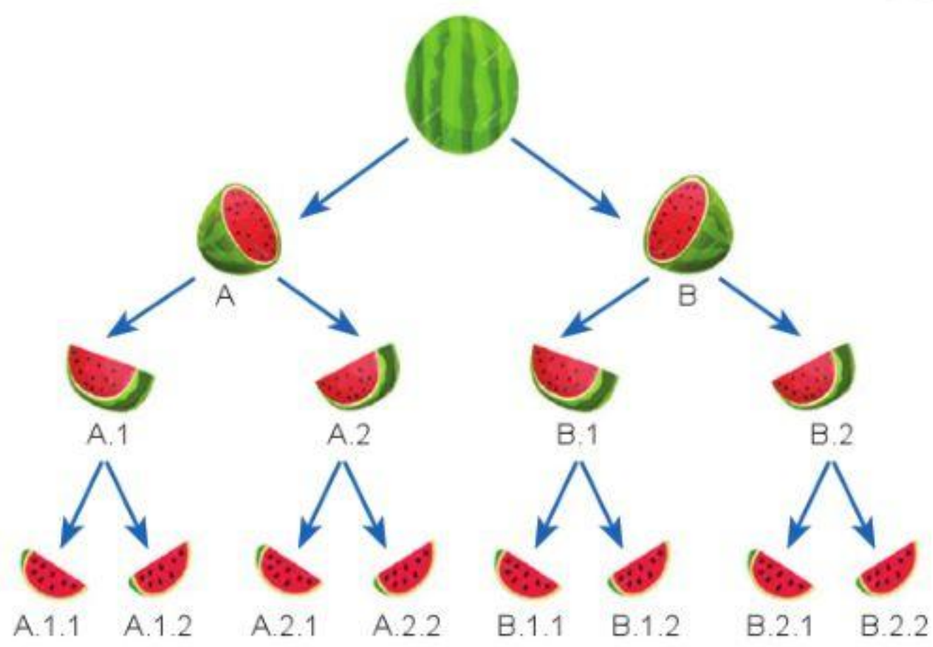
2. Xử lí từng miếng bằng cách chia nhỏ tiếp mỗi miếng dưa thành hai miếng nhỏ hơn. Quá trình này kết thúc khi đạt được số lượng miếng dưa mong muốn.

Cuối cùng, Thanh An đã có 8 miếng dưa hấu khá đều nhau và tiến hành đếm số hạt của mỗi miếng dưa này. Sau đó, Thanh An cộng dồn số hạt của hai miếng dưa ở mỗi lượt bổ theo thứ tự 3, 2, 1 để biết tổng số hạt của quả dưa hấu (*Hình 2*):

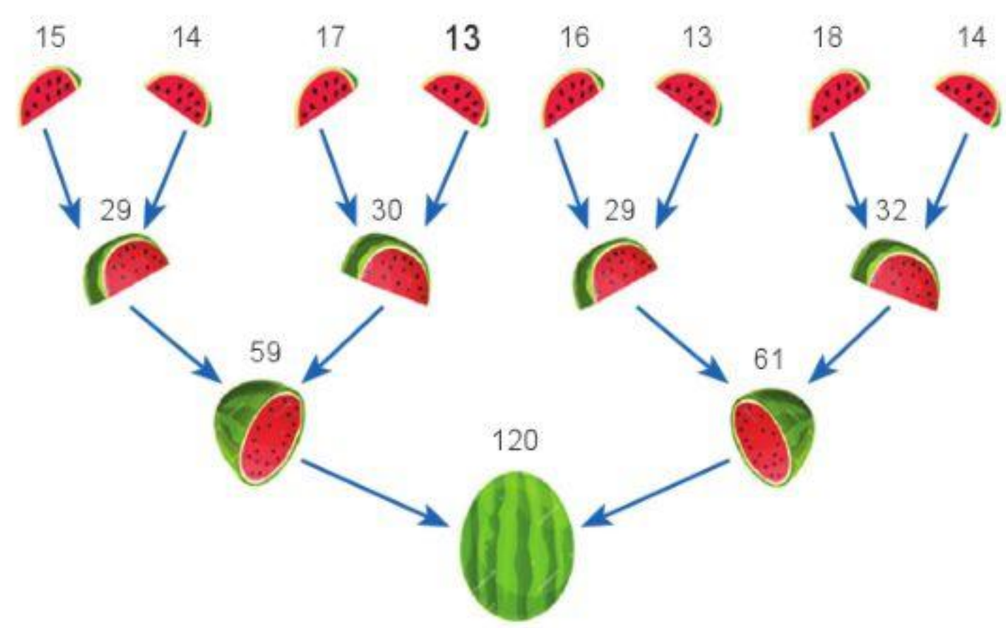
Lượt 1: Cộng dồn số hạt của từng cặp miếng dưa (A.1.1) và (A.1.2), (A.2.1) và (A.2.2), (B.1.1) và (B.1.2), (B.2.1) và (B.2.2) để biết số hạt của các miếng (A.1), (A.2), (B.1), (B.2).

Lượt 2: Cộng dồn số hạt của từng cặp miếng dưa (A.1) và (A.2), (B.1) và (B.2) để biết số hạt của các miếng (A) và (B).

Lượt 3: Cộng dồn số hạt của hai miếng dưa (A) và (B) để biết tổng số hạt của quả dưa.



Hình 1. Minh hoạ quá trình bổ các miếng dưa làm đôi



Hình 2. Minh hoạ quá trình cộng dồn số hạt của các miếng dưa

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

Nhận xét: Việc cộng dồn số hạt của các miếng dưa ở lượt sau giúp Thanh An biết được số hạt của mỗi miếng dưa ở lượt trước và tổng số hạt của quả dưa. Ở đây, Thanh An bổ đôi từng miếng dưa giúp giải quyết mục tiêu thứ nhất của bài toán là ước lượng hai nửa cho đều nhau. Tiếp theo, Thanh An cộng dồn số hạt mỗi miếng ngược theo thứ tự bổ giúp giải quyết mục tiêu thứ hai và thứ ba của bài toán là kiểm tra xem số hạt từng nửa chia có đều nhau hay không và đếm tổng số hạt trong quả dưa là bao nhiêu.

Cách giải quyết bài toán trên thể hiện ý tưởng chia để trị, bao gồm 3 bước:

1. Chia: Chia bài toán ban đầu (phức tạp) thành hai hoặc nhiều bài toán con (đơn giản hơn). Tiếp tục chia mỗi bài toán con thành các bài toán con đơn giản hơn nữa và cứ như thế cho đến khi đạt được các bài toán con đủ đơn giản mà chúng được giải quyết một cách dễ dàng.

2. Trị: Giải quyết các bài toán con (một cách đệ quy), kết quả là các lời giải của các bài toán con.

3. Kết hợp: Kết hợp các lời giải của các bài toán con để có được lời giải của bài toán ban đầu.

2

Trong các bài toán tìm kiếm trên một không gian xác định, thu hẹp dần phạm vi tìm kiếm là một kĩ thuật của ý tưởng chia để trị. Em hãy tìm hiểu bài toán sau đây và cho biết ý tưởng chia để trị được thể hiện trong kĩ thuật thu hẹp phạm vi tìm kiếm.

Bài toán *Tìm địa điểm du lịch*

Thanh An là một thanh niên sống ở Hà Nội và rất thích đi du lịch khám phá. Để lên kế hoạch cho kì nghỉ lần này, Thanh An quan sát 25 tỉnh thành được liệt kê trên lược đồ một số địa điểm du lịch ở miền Bắc Việt Nam (*Hình 3*) để chọn địa điểm phù hợp. Nhằm hạn chế dần số lượng điểm tìm kiếm, Thanh An suy nghĩ lần lượt các tiêu chí khám phá như sau:

– Tiêu chí đầu tiên Thanh An nhìn vào 2 vùng của miền Bắc là vùng Đồng bằng sông Hồng, vùng Trung du và miền núi phía Bắc. Thanh An quyết định chọn trong vùng từ trước đến nay ít đi du lịch hơn là vùng Trung du và miền núi phía Bắc. Số lượng địa điểm cần quan tâm trên bản đồ chỉ còn 15 tỉnh.

– Tiêu chí thứ hai là chỉ chọn những địa điểm du lịch nổi tiếng đã được in đậm trên bản đồ. Số lượng địa điểm cần quan tâm trên bản đồ chỉ còn 5 tỉnh.

– Tiêu chí thứ ba là địa điểm du lịch xa Hà Nội nhất do có kì nghỉ dài ngày. Kết quả Thanh An tìm ra địa điểm phù hợp nhất với các tiêu chí đã đặt ra đó là Cột Cờ Lũng Cú tại tỉnh Hà Giang.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Hình 3. Lược đồ một số địa điểm du lịch ở miền Bắc Việt Nam

Nhận xét: Việc xét lần lượt các tiêu chí giúp Thanh An từng bước giảm bớt số lượng lớn địa điểm cần tìm, từ 25 tỉnh xuống 15 tỉnh rồi xuống 5 tỉnh và cuối cùng tìm ra được địa điểm phù hợp nhất. Việc giới hạn dần phạm vi tìm kiếm địa điểm như vậy giúp Thanh An tìm ra địa điểm dễ dàng hơn nhiều so với việc ghép lần lượt từng địa điểm trên cả bản đồ vào các tiêu chí của mình đặt ra. Cách làm này thể hiện hai đặc điểm:

1. Thu hẹp không gian tìm kiếm của bài toán để đưa về bài toán nhỏ hơn.

2. Giải quyết bài toán nhỏ bằng cách tiếp tục thu hẹp không gian tìm kiếm bài toán để đưa về bài toán nhỏ hơn cho đến khi đạt được kết quả cần tìm.

Thu hẹp dần phạm vi tìm kiếm là một kĩ thuật của chia để trị. Kĩ thuật này được áp dụng trong các bài toán có thể loại bỏ đi những phần không gian tìm kiếm mà chắc chắn nghiệm của bài toán không nằm trong đó để giảm bớt độ phức tạp tính toán của thuật toán.

Thực hành: Với bài toán tìm địa điểm du lịch, em hãy đảo thứ tự các tiêu chí, thay đổi một số tiêu chí tìm địa điểm du lịch và quan sát các kết quả trung gian để thu được kết quả cuối cùng.

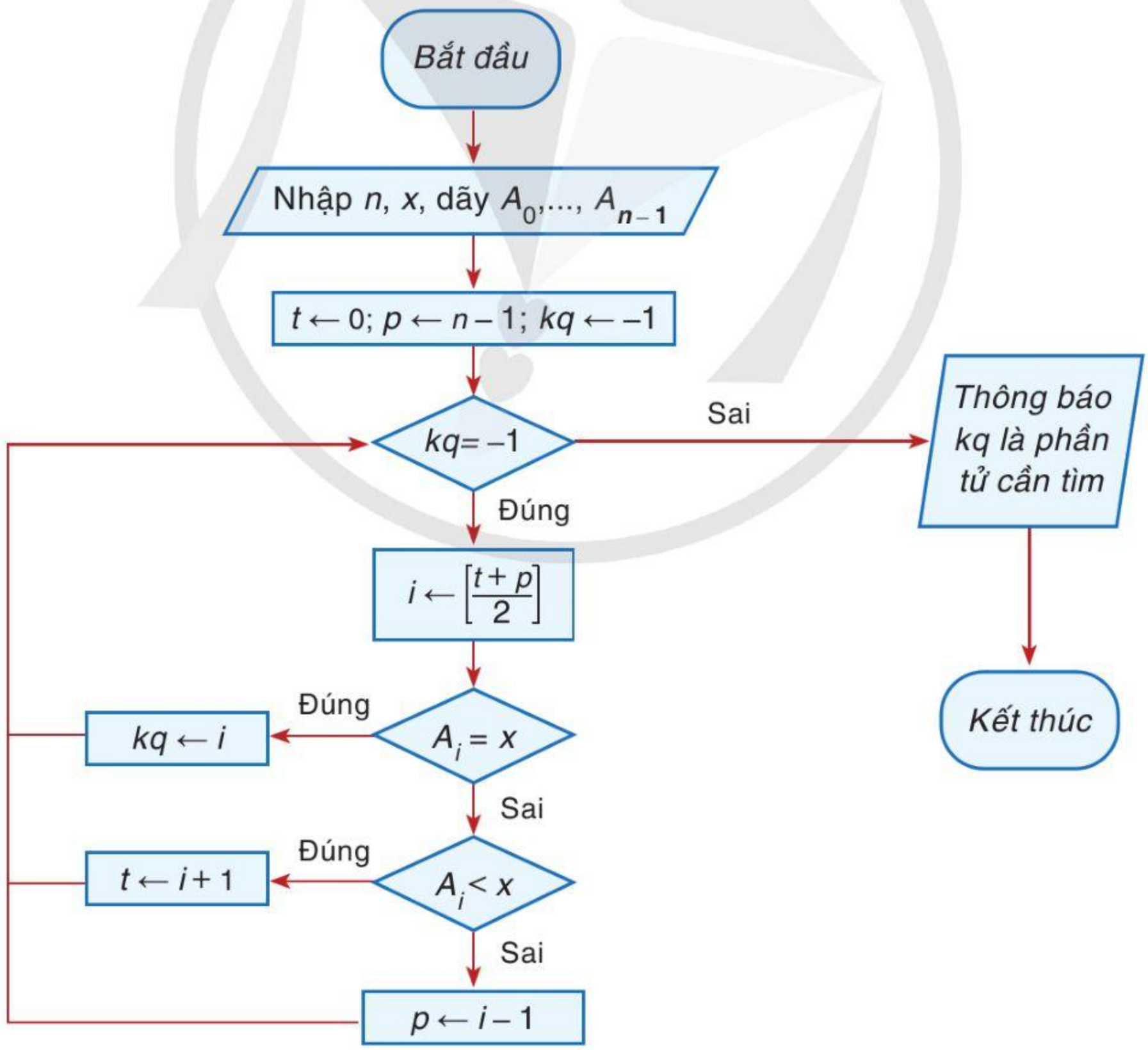
Đọc bản mới nhất trên hoc10.vn
Bản mẫu

2 Thuật toán *Tìm kiếm nhị phân*

Tìm kiếm nhị phân là một thuật toán cơ bản trong kĩ thuật thu hẹp phạm vi tìm kiếm của phương pháp chia để trị. Thuật toán tìm kiếm nhị phân trên dãy số sắp xếp tăng dần đã được giới thiệu ở sách *Tin học 7*. Ý tưởng của thuật toán là tìm kiếm một phần tử bằng cách chia dãy làm hai nửa, loại bỏ nửa dãy chắc chắn không chứa phần tử cần tìm, chỉ tìm kiếm trong nửa dãy còn lại. Phần này giới thiệu thuật toán tìm kiếm nhị phân dùng vòng lặp trên dãy số tăng dần bằng ngôn ngữ lập trình Python.

Bài toán 1. Cho dãy A gồm n phần tử phân biệt được sắp xếp theo thứ tự tăng dần của giá trị: $A_0 < A_1 < ... < A_{n-1}$. Em hãy lập trình thuật toán tìm kiếm nhị phân nhập vào hai số nguyên n, x và dãy số tăng dần $A_0, A_1, ..., A_{n-1}$; sau đó trả về một chỉ số i $(0 \le i \le n-1)$ sao cho $A_i = x$. Dữ liệu vào đảm bảo giá trị x luôn xuất hiện trong dãy A.

Thực hành: Quan sát sơ đồ khối ở *Hình 4* và đọc hiểu chương trình tìm kiếm nhị phân trên dãy số tăng dần trong *Hình 5*. Em hãy kiểm thử chương trình với các bộ dữ liệu thử nghiệm.



Hình 4. Sơ đồ khối mô tả thuật toán tìm kiếm nhị phân trên dãy số tăng dần

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

File Edit Format Run Options Window Help

```
n, x = map(int, input().split())
A = list(map(int, input().split()))

t = 0
p = n-1
kq = -1
while (kq == -1):

    #Xác định vị trí i giữa t và p
     i = (t + p) // 2

    #Kiểm tra nếu x bằng phần tử số i
    if A[i] == x:
        kq = i #Đạt điều kiện để dừng
               #vòng lặp

    #Nếu x lớn hơn, bỏ qua nửa bên trái
    elif A[i] < x:
         t = i + 1

    #Nếu x nhỏ hơn, bỏ qua nửa bên phải
    else:
         p = i - 1

print("Phần tử số",kq,"có giá trị",x)
```

```
7 10
4 7 8 10 14 21 22
Phần tử số 3 có giá trị 10
```

```
7 22
4 7 8 10 14 21 22
Phần tử số 6 có giá trị 22
```

Ghi chú

- `n, x = map(int, input().split())`: tách dòng dữ liệu đọc vào thành các số và gán tương ứng vào hai biến `n`, `x`.
- `A = list(map(int, input().split()))`: tách dòng dữ liệu đọc vào thành các số và tạo thành danh sách sau đó gán vào danh sách `A`.

Hình 5. Chương trình tìm kiếm nhị phân trên dãy số tăng dần và màn hình kết quả chạy một số bộ dữ liệu thử nghiệm

Nếu bỏ qua điều kiện giá trị x bắt buộc phải xuất hiện trong dãy số ở bài toán trên, ta có bài toán sau đây.

Bài toán 2. Cho dãy A gồm n phần tử phân biệt sắp xếp theo thứ tự tăng dần của giá trị: $A_0 < A_1 < \ldots < A_{n-1}$. Hãy lập trình thuật toán tìm kiếm nhị phân nhập vào hai số tự nhiên n, x cùng với dãy số tăng dần $A_0, A_1, \ldots, A_{n-1}$; nếu x là một giá trị xuất hiện trong dãy, đưa ra một chỉ số i ($0 \le i \le n-1$) sao cho $A_i = x$; nếu không, đưa ra thông báo không tồn tại giá trị x trong dãy.

Cách giải bài toán này tương tự với cách giải của Bài toán 1 và thay đổi công thức tính chỉ số i. Em hãy sửa lại chương trình trong *Hình 5* để giải quyết thêm trường hợp không tìm thấy x trong dãy.

Thực hành: Quan sát chương trình trong *Hình 6* đã được chỉnh sửa sau và cho biết:

1) Dấu ? cần được thay thế bởi phép toán gì để có thể hiển thị thông báo không tìm thấy x trong dãy.

2) Kiểm thử chương trình với các bộ dữ liệu thử nghiệm kích thước nhỏ.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

File Edit Format Run Options Window Help

```python
n, x = map(int, input().split())
A = list(map(int, input().split()))
t = 0
p = n-1
kq = -1
while (t <= p) and (kq == -1):
    i = t + (p - t) // 2
    #Nếu x lớn hơn, tìm x trong nửa dãy bên phải
    if A[i] == x:
        kq = i # Đạt điều kiện kết thúc vòng lặp
    #Nếu x lớn hơn, bỏ qua nửa bên trái
    elif A[i] < x:
        t = i + 1
    #Nếu x nhỏ hơn, tìm x trong nửa dãy bên trái
    else:
        p = i - 1
if t > p:
    print("Không tồn tại phần tử có giá trị",x)
else:
    print("Phần tử số",kq,"có giá trị",x)
```

```
6 21
4 7 8 10 14 21
Phần tử số 5 có giá trị 21
```

```
6 8
4 7 8 10 14 21
Phần tử số 2 có giá trị 8
```

```
6 11
4 7 8 10 14 21
Không tồn tại phần tử có
giá trị 11
```

Hình 6. Chương trình tìm kiếm nhị phân trên dãy số tăng dần cho bài thực hành và màn hình kết quả chạy một số bộ dữ liệu thử nghiệm

Em hãy viết chương trình tìm kiếm nhị phân giá trị x trong dãy số không giảm A có n phần tử, các phần tử có thể trùng nhau; kết quả là hiển thị chỉ số nhỏ nhất i sao cho $A_i = x$ hoặc hiển thị thông báo không tìm thấy x.

Câu 1. Trong thuật toán tìm kiếm nhị phân, tìm một phần tử có giá trị x trong dãy số có 20 phần tử, em hãy cho biết sau hai bước lặp chia đôi để tìm kiếm mà vẫn chưa tìm được giá trị x đó thì độ lớn không gian tìm kiếm còn lại (tức là độ dài đoạn dãy số cần tìm) là bao nhiêu.

A. 2 B. 4 C. 5 D. 8

Câu 2. Hai công thức tính chỉ số i trong hai chương trình của *Hình 5* và *Hình 6* có khác nhau. Em hãy cho biết hai chương trình này có cùng kết quả tìm kiếm không.

Tóm tắt bài học

- Ý tưởng của chia để trị: Chia bài toán ban đầu (phức tạp) thành hai hoặc nhiều bài toán con (đơn giản hơn). Sau đó, giải quyết các bài toán con và có được các lời giải của các bài toán con. Cuối cùng, kết hợp các lời giải của các bài toán con để có được lời giải của bài toán ban đầu.
- Tìm kiếm nhị phân là một thuật toán của chia để trị để thu hẹp phạm vi tìm kiếm. Tại mỗi bước lặp, phạm vi tìm kiếm bị giảm một nửa.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bài 2

KĨ THUẬT ĐỆ QUY TRONG CHIA ĐỂ TRỊ

Học xong bài này, em sẽ:

- Nêu được ý tưởng kĩ thuật đệ quy trong bài toán tìm kiếm nhị phân và tính a^n.
- Biết được một số bài toán sử dụng kĩ thuật đệ quy trong chia để trị.
- Cài đặt được thuật toán tìm kiếm nhị phân bằng kĩ thuật đệ quy.
- Hiểu và mô tả được thuật toán giải bài toán tính a^n.

Trong Bài 1, em đã biết thuật toán tìm kiếm nhị phân bằng vòng lặp. Việc loại bỏ đi một nửa dãy sau mỗi bước và tìm kiếm phần tử trên một nửa dãy còn lại cũng phù hợp với việc cài đặt đệ quy do các bước làm chỉ khác nhau ở phạm vi tìm kiếm. Em hãy mô tả lại từng bước thu hẹp phạm vi tìm kiếm trên một ví dụ trong *Hình 4* của Bài 1 để thấy sự lặp lại thuật toán trên bài toán con so với bài toán cha.

1 Cài đặt thuật toán tìm kiếm nhị phân bằng đệ quy

Thuật toán tìm kiếm nhị phân được mô tả chi tiết từng bước thực hiện như sau:

Bước 1. So sánh x với phần tử nằm ở vị trí giữa dãy số, gọi là phần tử giữa.

Bước 2. Nếu x bằng với giá trị phần tử giữa, đưa ra vị trí phần tử tìm được.

Bước 3. Nếu x lớn hơn giá trị phần tử giữa, giá trị x chỉ có thể nằm ở nửa bên phải phần tử giữa của dãy số (nửa có giá trị lớn hơn). Quay lại *Bước 1*, tiếp tục áp dụng thuật toán đối với nửa dãy số bên phải này.

Bước 4. Nếu x nhỏ hơn giá trị phần tử giữa, giá trị x chỉ có thể nằm ở nửa bên trái phần tử giữa của dãy số (nửa có giá trị nhỏ hơn). Quay lại *Bước 1*, tiếp tục áp dụng thuật toán đối với nửa dãy số bên trái này.

1

Tìm hiểu *Bước 3* và *Bước 4* trong thuật toán tìm kiếm nhị phân để rút ra kĩ thuật đệ quy cài đặt thuật toán này. Hai bước trên có thể cài đặt bởi lời gọi đệ quy đến hàm tìm kiếm nhị phân tổng quát với tham số đầu vào là khoảng cần tìm kiếm trong dãy số. Em hãy đọc hiểu chương trình Python mẫu trong *Hình 1* và chạy thử nghiệm trên các bộ dữ liệu đầu vào khác nhau.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

***Nhận xét*:** Chức năng hoạt động của *Bước 3* và *Bước 4* trong thuật toán tìm kiếm nhị phân hoàn toàn giống nhau chỉ khác là thực hiện ở phạm vi nào trong dãy số. Đồng thời chức năng hoạt động của hai bước này cũng hoàn toàn giống với chức năng hoạt động của thuật toán trên toàn dãy số. Lưu ý trong trường hợp phạm vi tìm kiếm là rỗng (nghĩa là $t > p$ trong chương trình ở *Hình 1*), chương trình cần thông báo không tồn tại phần tử cần tìm.

File Edit Format Run Options Window Help

```
1  def TKNP(A, t, p, x):
2      if t <= p:
3
4          #Xác định vị trí i giữa t và p
5          i = (t + p) // 2
6
7          #Kiểm tra nếu x bằng phần tử số i
8          if A[i] == x:
9              return i
10
12         #Nếu x nhỏ hơn, bỏ qua nửa bên phải,
13         #gọi đệ quy TKNP cho phần bên trái
14         elif A[i] > x:
15             return TKNP(A, t, i-1, x)
16
17         #Nếu x lớn hơn, bỏ qua nửa bên trái,
18         #gọi đệ quy TKNP cho phần bên phải
19         else:
20             return TKNP(A, i + 1, p, x)
21
22     else:
23         #Không tồn tại phần tử cần tìm trong mảng
24         return -1
25
26 n, x = map(int, input().split())
27 A = list(map(int, input().split()))
28
29 #Gọi đệ quy
30 kq = TKNP(A, 0, len(A)-1, x)
31
32 if kq == -1:
33     print("Không tồn tại phần tử có giá trị",x)
34 else:
35     print("Phần tử số",kq,"có giá trị",x)
```

```
6 21
4 7 8 10 14 21
Phần tử số 5 có giá trị 21
```

```
6 8
4 7 8 10 14 21
Phần tử số 2 có giá trị 8
```

```
6 11
4 7 8 10 14 21
Không tồn tại phần tử có
giá trị 11
```

Hình 1. Chương trình đệ quy để tìm kiếm nhị phân trên dãy số không giảm cho bài thực hành và màn hình kết quả chạy một số bộ dữ liệu thử nghiệm

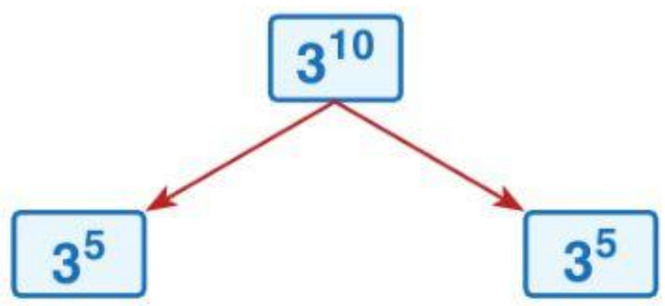
2 Bài toán *Tính a^n*

Trong giờ học môn Toán, thầy giáo ra câu hỏi cho cả lớp xem ai tính nhanh nhất giá trị của 3^{10}. Thanh An nghĩ rằng: Nếu cứ nhân lần lượt 10 số 3 với nhau thì sẽ phải mất 9 phép nhân. Do mới học chuyên đề Khoa học máy tính phần Chia để trị,

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Thanh An nghĩ ra cách để giảm bớt số phép nhân. Thanh An thấy rằng $3^{10} = 3^5 \times 3^5$ (*Hình 2*) nên chỉ cần một lần tính 3^5 và bình phương giá trị đó lên thì số phép nhân phải sử dụng giảm đi đáng kể. Tuy nhiên, với 3^5 thì Thanh An lại chưa nghĩ ra được cách chia đôi ra để giảm bớt số phép nhân do 5 là số lẻ.

Khi về nhà, Thanh An mong muốn lập trình giải quyết bài toán tổng quát tính a^n với a và n là hai số nguyên dương.

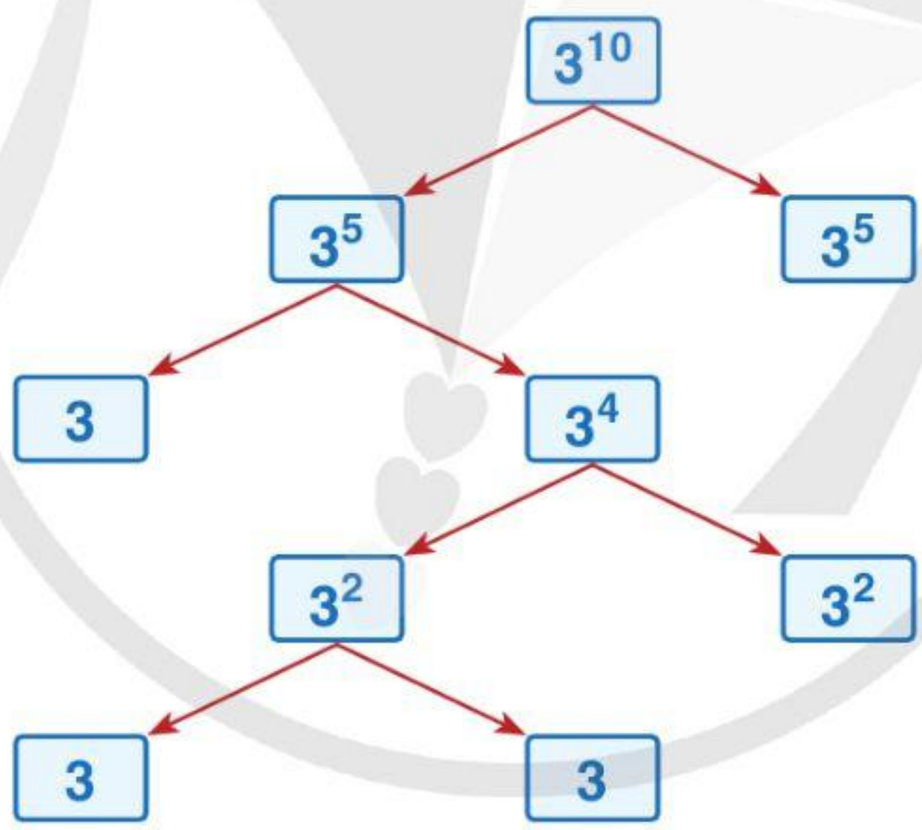


Hình 2. Cách tính $3^{10} = 3^5 \times 3^5$ dùng chia để trị

2

Em hãy giúp Thanh An mô tả chi tiết các bước tính giá trị 3^{10} với số phép tính nhân phải sử dụng là ít nhất.

Gợi ý cách làm giúp Thanh An qua sơ đồ *Hình 3*:



Hình 3. Cách tính 3^{10} dùng chia để trị có số phép tính nhân là ít nhất

Thực hành: Mô tả chi tiết tính a^n dùng chia để trị mà sử dụng số phép tính nhân là ít nhất. Viết hàm đệ quy tính a^n. Sau đó, viết chương trình nhập vào hai số nguyên dương a, n và hiển thị giá trị của a^n.

Hướng dẫn: Có hai trường hợp:

1) $a^n = a^{\frac{n}{2}} \times a^{\frac{n}{2}}$, nếu n chẵn.

2) $a^n = a \times a^{n-1}$, nếu n lẻ và $n > 1$.

Lưu ý: Lưu lại kết quả bài toán con trong mỗi lần gọi đệ quy để tránh gọi đệ quy lặp lại cho các bài toán con đã tính.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Kiểm thử chương trình:

Em hãy chạy kiểm thử chương trình của phần thực hành với các bộ dữ liệu kiểm thử trong *Bảng 1*.

Nếu chương trình cho kết quả sai với một bộ dữ liệu kiểm thử, em hãy thêm vào các lệnh để in ra giá trị của các biến; sau đó, chạy lại chương trình với bộ dữ liệu kiểm thử này, theo dõi sự thay đổi giá trị của các biến và phát hiện lệnh nào tính toán sai.

Bảng 1. Một số bộ dữ liệu thử nghiệm cho bài toán a^n

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	2 10	2 mũ 10 bằng 1024
2	4 7	4 mũ 7 bằng 16384
3	3 10	3 mũ 10 bằng 59049

Em hãy viết hàm đệ quy để tìm kiếm nhị phân giá trị x trong dãy A không giảm có n phần tử $A_0, A_1, \ldots, A_{n-1}$, các phần tử có thể trùng nhau. Nếu tìm thấy thì hàm này trả về chỉ số i nhỏ nhất mà $A_i = x$. Nếu không tìm thấy thì hàm này trả về -1.

Câu 1. Trong những câu sau đây, câu nào đúng cho việc giải bài toán tính a^n bằng phương pháp chia để trị?

1) Xét trường hợp n chẵn và n lẻ riêng.

2) n chẵn hay n lẻ đều giải quyết như nhau.

Câu 2. Em hãy cho biết, nếu sử dụng phương pháp chia để trị để tính 4^{12} thì cần ít nhất bao nhiêu phép tính nhân.

A. 4 B. 5 C. 6 D. 7

Tóm tắt bài học

- Thuật toán tìm kiếm nhị phân phù hợp để cài đặt bằng kĩ thuật đệ quy do việc học lại chức năng thuật toán của các bài toán con.
- Bài toán tính a^n được giải hiệu quả bằng kĩ thuật đệ quy trong phương pháp chia để trị nhờ việc xét trường hợp n chẵn và n lẻ.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bài 3

THỰC HÀNH ỨNG DỤNG THUẬT TOÁN TÌM KIẾM NHỊ PHÂN BẰNG ĐỆ QUY

Học xong bài này, em sẽ:

- Áp dụng được phương pháp tìm kiếm nhị phân vào bài toán tìm phần tử lớn nhất trong mảng có phần đầu sắp xếp tăng dần và phần sau sắp xếp giảm dần.
- Viết được chương trình để giải một bài toán dùng kĩ thuật chia để trị bằng cách thu hẹp phạm vi tìm kiếm.

Bài toán *Tìm phần tử lớn nhất trong mảng có phần đầu sắp xếp tăng dần và phần sau sắp xếp giảm dần*

Sau giờ chào cờ, trường của Thanh An tổ chức hoạt động kết nối ở sân trường. Lớp của Thanh An được xếp thành một hàng theo chiều cao tăng dần. Lớp của Hải Bình được xếp thành một hàng theo chiều cao giảm dần. Sau đó, nhập hai lớp này thành một hàng bằng cách bạn đầu tiên của lớp Hải Bình đứng vào sau bạn cuối cùng của lớp Thanh An và tạo thành một hàng như *Hình 1*. Thầy giáo phụ trách yêu cầu đưa ra được cách tìm bạn có chiều cao lớn nhất trong hàng mà sử dụng ít phép so sánh nhất.

Hình 1. Minh hoạ hai lớp xếp hàng

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

Yêu cầu: Cho dãy A gồm n phần tử có giá trị đôi một khác nhau $A_0, A_1, \ldots, A_{n-1}$ sao cho tồn tại $k\,(0 < k < n-1)$ để $A_0 < A_1 < \ldots < A_k$ và $A_k > A_{k+1} > \ldots > A_{n-1}$. Em hãy viết chương trình tìm phần tử số k sao cho số lần so sánh là ít nhất.

Lưu ý: A_k là phần tử có giá trị lớn nhất của dãy A.

Để giải quyết bài toán trên, hãy thực hiện lần lượt các bài thực hành sau.

Thực hành 1: Mô tả chi tiết cách giải bài toán trên dùng phương pháp tìm kiếm nhị phân.

Hướng dẫn: Các bước bao gồm:

Bước 1 (Chia). Xác định vị trí k ở chính giữa dãy A.

Bước 2 (Trị). Xác định dãy bên trái hay bên phải của A_k chứa phần tử có giá trị lớn nhất trong dãy, quay trở lại *Bước 1* tiếp tục tìm trên dãy mới đó. Quá trình kết thúc khi xác định được phần tử có giá trị lớn nhất.

Thực hành 2: Viết chương trình dùng đệ quy: Nhập vào giá trị n và n giá trị $A_0, A_1, \ldots, A_{n-1}$ có dạng phần đầu giá trị tăng dần và phần sau giá trị giảm dần, hãy hiển thị phần tử có giá trị lớn nhất của dãy A.

Hướng dẫn: Em hãy dựa vào cách viết chương trình tìm kiếm nhị phân để viết chương trình cho bài toán này.

Kiểm thử chương trình:

Em hãy chạy kiểm thử chương trình của phần thực hành với các dữ liệu kiểm thử trong *Bảng 1*.

Nếu chương trình cho kết quả sai với một bộ dữ liệu kiểm thử thì thêm vào các lệnh để in ra giá trị của các biến; sau đó, chạy lại chương trình với bộ dữ liệu kiểm thử này, theo dõi sự thay đổi giá trị của các biến và phát hiện lệnh nào tính toán sai.

Bảng 1. Một số bộ dữ liệu thử nghiệm cho Thực hành 2

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	12 4 7 8 10 14 21 22 36 91 81 77 62	Phần tử lớn nhất trong dãy có giá trị 91 ở vị trí số 8
2	12 4 7 8 10 91 81 77 62 36 22 21 14	Phần tử lớn nhất trong dãy có giá trị 91 ở vị trí số 4
3	10 2 4 6 8 10 9 7 5 3 1	Phần tử lớn nhất trong dãy có giá trị 10 ở vị trí số 4

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Thực hành 3: Viết chương trình tìm kiếm tuần tự cho bài toán trên. Với mỗi bộ dữ liệu thử nghiệm, em hãy so sánh số lần lặp của chương trình tìm kiếm tuần tự (dùng vòng lặp) với số lần gọi đệ quy của chương trình ở phần Thực hành 2 (dùng đệ quy). Từ đó, với nhiều bộ dữ liệu thử nghiệm, em sẽ nhận thấy phương pháp tìm kiếm nhị phân có số lần lặp ít hơn nhiều so với phương pháp tìm kiếm tuần tự.

Hướng dẫn:

Các bước bao gồm:

Bước 1. Viết chương trình tìm kiếm tuần tự dùng vòng lặp để tìm phần tử A_k. Sử dụng một biến đếm để đếm số lần lặp.

Bước 2. Sử dụng một biến đếm để đếm số lần thực hiện hàm đệ quy của phần Thực hành 2.

Cho dãy A gồm n phần tử $A_0, A_1, \ldots, A_{n-1}$ sao cho tồn tại k $(0 < k < n - 1)$ để $A_0 \leq A_1 \leq \ldots \leq A_k$ và $A_k \geq A_{k+1} \geq \ldots \geq A_{n-1}$. Em hãy viết chương trình dùng đệ quy để hiển thị chỉ số i là chỉ số nhỏ nhất mà A_i có giá trị lớn nhất của dãy A. Em hãy chạy kiểm thử chương trình này với các bộ dữ liệu thử nghiệm trong *Bảng 2*.

Bảng 2. Một số bộ dữ liệu thử nghiệm cho bài toán Vận dụng

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	12 4 7 8 8 14 21 22 22 91 81 77 62	Phần tử lớn nhất trong dãy có giá trị 91 ở vị trí số 8
2	12 4 7 7 7 91 81 81 62 36 22 21 14	Phần tử lớn nhất trong dãy có giá trị 91 ở vị trí số 4
3	10 2 2 6 10 10 11 9 5 3 1	Phần tử lớn nhất trong dãy có giá trị 11 ở vị trí số 5
4	9 31 4 7 12 3 60 88 32 20	Phần tử lớn nhất trong dãy có giá trị 88 ở vị trí số 6
5	11 18 41 10 7 28 40 51 16 99 63 71	Phần tử lớn nhất trong dãy có giá trị 99 ở vị trí số 8

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bài 4

KĨ THUẬT CHIA ĐỂ TRỊ TRONG THUẬT TOÁN SẮP XẾP TRỘN

Học xong bài này, em sẽ:

- Hiểu được các bước cần thực hiện khi giải bài toán sắp xếp dãy số bằng thuật toán sắp xếp trộn.
- Biết được cách cài đặt thuật toán sắp xếp trộn.
- Nêu được ý tưởng kĩ thuật chia để trị tổng quát bằng đệ quy.
- Biết được các bước cơ bản của kĩ thuật chia để trị để giải một bài toán.

Bài 2 giới thiệu kĩ thuật đệ quy trong phương pháp chia để trị. Nhiều bài được giải quyết dễ dàng bằng cách sử dụng kĩ thuật đệ quy. Ví dụ: Em hãy chia đôi dãy gồm bốn số {7, 3, 8, 2} làm hai nửa để thực hiện công việc sắp xếp bốn số này theo thứ tự tăng dần của giá trị.

Gợi ý: 7 3 8 2 → 3 7 2 8 → 2 3 7 8

Một ví dụ điển hình của phương pháp chia để trị là thuật toán sắp xếp trộn dùng kĩ thuật đệ quy.

1 Ý tưởng thuật toán sắp xếp trộn

Thầy giáo lớp Thanh An mời 5 bạn học sinh lên bảng, xếp ngẫu nhiên thành một hàng ngang, từ trái sang phải là các bạn có tên Thi, An, Hoà, Lâm, Mai. Em hãy giúp thầy giáo yêu cầu 5 bạn thực hiện lần lượt các bước trong hai giai đoạn sau để sắp xếp hàng tăng dần theo chiều cao từ trái sang phải.

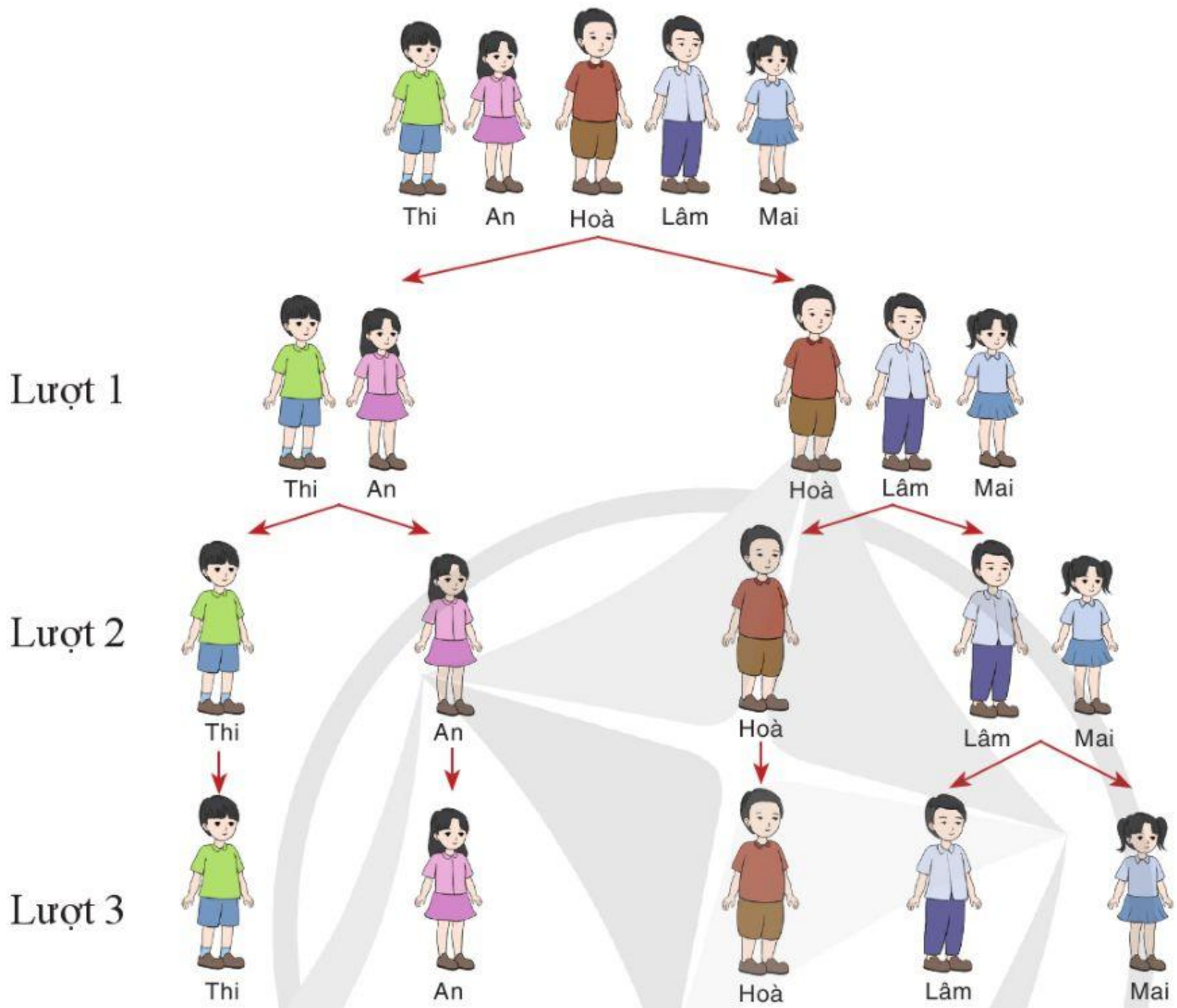
Giai đoạn 1: Ở giai đoạn này, với mỗi lượt, mỗi dãy được chia làm hai dãy nhỏ với mục tiêu sắp xếp tăng dần trên từng dãy nhỏ (*Hình 1*). Quá trình kết thúc khi mỗi dãy có đúng một bạn:

Lượt 1: Chia thành hai dãy, một dãy 2 bạn (Thi, An) và một dãy 3 bạn (Hoà, Lâm, Mai).

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Lượt 2: Chia mỗi dãy trong hai dãy trên thành hai dãy nhỏ, lần lượt là: (Thi), (An); (Hoà) và (Lâm, Mai).

Lượt 3: Chia dãy (Lâm, Mai) thành hai dãy mỗi dãy có đúng một bạn: (Lâm), (Mai).



Hình 1. Minh hoạ Giai đoạn 1

Giai đoạn 2: Ở giai đoạn này, với mỗi lượt, cứ hai dãy liên tiếp được tách ra từ Giai đoạn 1 theo trình tự lượt ngược lại 3, 2, 1 sẽ được kết hợp lại thành một dãy (*Hình 2*). Sau mỗi lượt, các bạn trong từng dãy nhỏ có chiều cao tăng dần:

Lượt 1: Kết hợp hai dãy (Lâm) và (Mai), Lâm cao hơn Mai, xếp Mai trước Lâm. Dãy trở thành (Mai, Lâm) có chiều cao tăng dần.

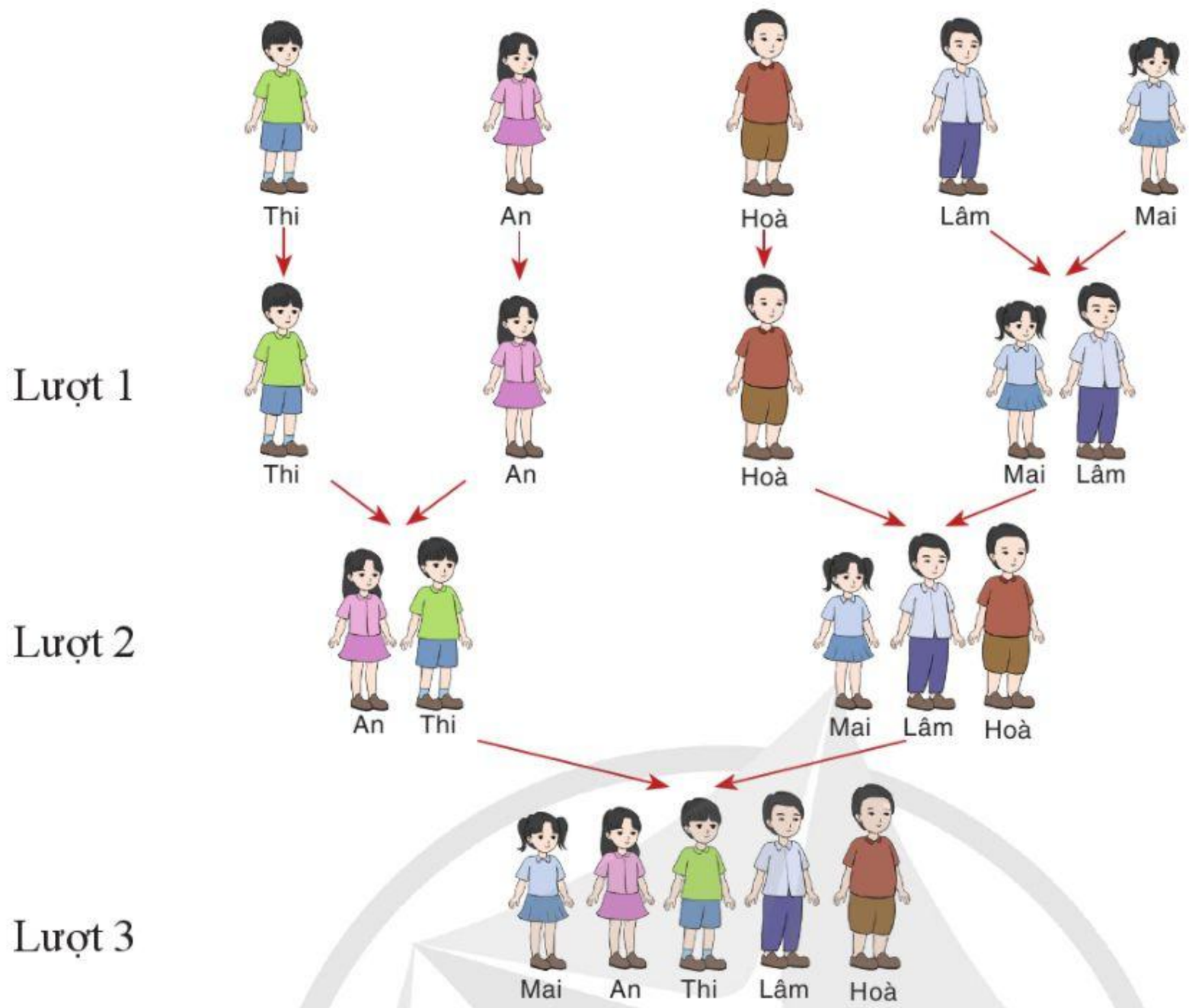
Lượt 2:

– Kết hợp hai dãy (Thi) và (An), Thi cao hơn An, xếp An trước Thi. Dãy trở thành (An, Thi) có chiều cao tăng dần.

– Kết hợp hai dãy (Hoà) và (Mai, Lâm), Hoà cao hơn Mai, xếp Mai vào đầu dãy; Hoà cao hơn Lâm, xếp Lâm đứng cạnh Mai và Hoà đứng cạnh Lâm. Nhóm trở thành (Mai, Lâm, Hoà) có chiều cao tăng dần.

Lượt 3: Kết hợp hai dãy (An, Thi) và (Mai, Lâm, Hoà), An cao hơn Mai, xếp Mai vào đầu dãy; An thấp hơn Lâm, xếp An đứng cạnh Mai; Thi thấp hơn Lâm, xếp Thi đứng cạnh An, sau đó đến Lâm và Hoà. Dãy trở thành (Mai, An, Thi, Lâm, Hoà) có chiều cao tăng dần.

Đọc bản mới nhất trên hoc10.vn
Bản mẫu



Hình 2. Minh hoạ Giai đoạn 2

Nhận xét: Thầy giáo lớp Thanh An chia dần 5 bạn thành các dãy nhỏ giúp việc sắp xếp vị trí các bạn tăng dần theo chiều cao trở nên thuận lợi và dễ dàng. Cách làm trên của thầy giáo thể hiện ba bước cơ bản của kĩ thuật chia để trị cho hoạt động trên như sau:

1. Chia: Chia một dãy thành hai dãy nhỏ.

2. Trị: Sắp xếp mỗi dãy nhỏ theo cách giống như sắp xếp dãy ban đầu. Quá trình này kết thúc khi mỗi dãy nhỏ chỉ còn một bạn. Mục tiêu mỗi lượt ở Giai đoạn 1 là chia mỗi dãy thành hai dãy nhỏ và thực hiện lặp lại bài toán sắp xếp trên từng dãy nhỏ. Sau mỗi lượt ở Giai đoạn 2, mỗi dãy nhỏ đều được sắp xếp tăng dần và là kết quả của mục tiêu từng lượt ở Giai đoạn 1.

3. Kết hợp: Cứ hai dãy liên tiếp đã được sắp xếp kết hợp lại thành một dãy các bạn được sắp xếp theo chiều cao tăng dần. Bước này được làm lần lượt từ các dãy kích thước nhỏ đến lớn.

Sau đây, em sẽ được tìm hiểu thuật toán sắp xếp trộn để sắp xếp tăng dần một dãy số.

2 Thuật toán sắp xếp trộn

Bài toán: Cho dãy số A gồm n phần tử $A_0, A_1, \ldots, A_{n-1}$. Em hãy viết chương trình nhập vào số nguyên n và dãy số nguyên A có n phần tử $A_0, A_1, \ldots, A_{n-1}$; sau đó sắp xếp dãy này theo thứ tự tăng dần dùng thuật toán sắp xếp trộn.

Các bước của thuật toán sắp xếp trộn được thiết kế theo kĩ thuật đệ quy cho hàm sắp xếp trộn `Merge_sort(A)` như sau:

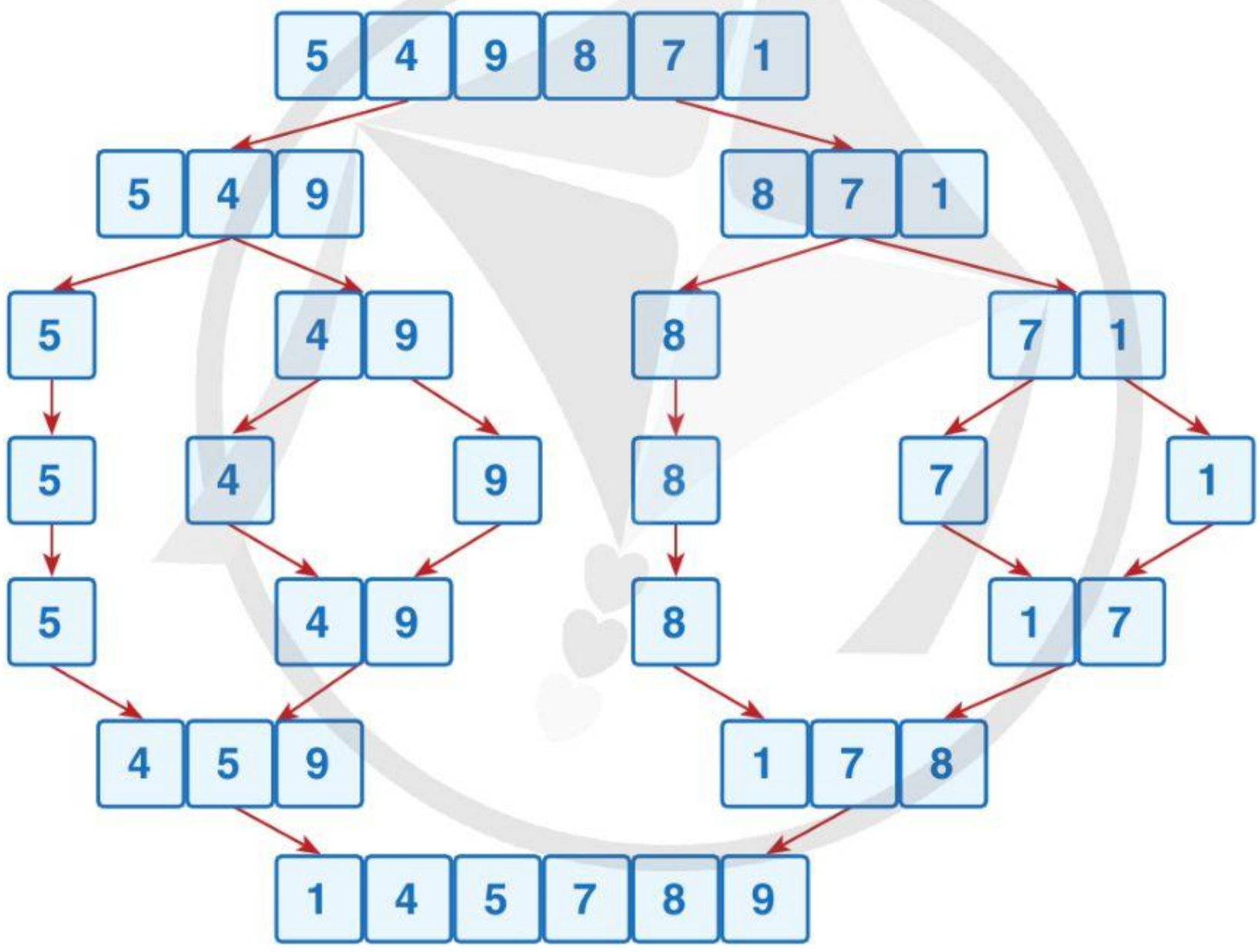
Đọc bản mới nhất trên hoc10.vn Bản mẫu

1. Chia: Chia dãy A thành hai dãy con $(A_0, A_1, \ldots, A_k)$ và $(A_k, A_{k+1}, \ldots, A_{n-1})$ có số lượng phần tử chênh nhau không quá 1.

2. Trị: Sắp xếp mỗi dãy con bằng cách sử dụng lời gọi đệ quy `Merge_sort()` với đầu vào mới tương ứng cho mỗi dãy con cần sắp xếp. Hàm đệ quy dừng khi độ dài dãy cần sắp xếp bằng 1.

3. Kết hợp: Trộn hai dãy con đã được sắp xếp lại thành một dãy duy nhất được sắp xếp tăng dần. Thuật toán kết hợp duy trì ba biến chạy, hai biến chạy cho hai dãy con, một biến chạy cho dãy được sắp xếp cuối cùng, tăng các biến chạy tương ứng với các vị trí trong dãy lần lượt từ trái qua phải, lấy ra phần tử nhỏ hơn trong hai giá trị tại vị trí hai biến chạy của hai dãy con và gán vào vị trí biến chạy của dãy sắp xếp cuối cùng.

Hình 3 là một ví dụ cho sắp xếp dãy số: 5, 4, 9, 8, 7, 1 dùng thuật toán trên.



Hình 3. Một ví dụ hoạt động của thuật toán sắp xếp trộn

Thực hành: Em hãy tìm hiểu các đoạn chương trình viết trên ngôn ngữ lập trình Python cho thuật toán sắp xếp trộn sau đây. Em hãy soạn thảo và chạy chương trình với các bộ dữ liệu thử nghiệm trong *Bảng 1*.

Trong *Hình 4*, hàm đệ quy sắp xếp trộn `Merge_sort(A)` dùng để sắp xếp trộn dãy A theo thứ tự tăng dần. Phần cơ sở là dãy A chỉ có một phần tử (xem như có thứ tự); do đó, phần cơ sở là rỗng. Hàm chỉ có phần đệ quy và quá trình đệ quy kết thúc khi dãy A chỉ có một phần tử (kết thúc bước *Chia*).

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

```
File Edit Format Run Options Window Help
1  def Merge_sort(A):
2      if len(A) <= 1: return #Phần cơ sở đệ quy
3      g = len(A)//2 #Xác định phần tử giữa của
4                    #dãy A
5      T = A[:g] #Sao chép vào dãy T nửa bên
6                #trái của A
7      P = A[g:] #Sao chép vào dãy P nửa bên
8                #phải của A
9      Merge_sort(T) #Gọi đệ quy sắp xếp nửa
10                   #bên trái
12     Merge_sort(P) #Gọi đệ quy sắp xếp nửa
13                   #bên phải
14     Merge(A,T,P) #Trộn 2 dãy con T, P đã được
15                  #sắp xếp thành dãy A sắp xếp
16                  #tăng dần
17 A = [9, 11, 44, 7, 2, 8, 11, 17, 31, 7]
18 Merge_sort(A)
19 print("Dãy sau khi sắp xếp là: ")
20 for i in range(len(A)):
21     print(A[i], end=" ")
```

Ghi chú

`end=" "` trong câu lệnh `print(A[i], end=" ")` có tác dụng ghi ra dấu cách thay vì xuống dòng.

Hình 4. Chương trình sắp xếp trộn mảng A với hàm đệ quy **`Merge_sort()`**

Trong *Hình 5*, hàm **`Merge(A,T,P)`** dùng để trộn hai dãy *T* và *P* vào dãy *A*. Dãy *T* và *P* đã được sắp xếp khi truyền vào hàm. Sau khi kết thúc hàm **`Merge()`**, các phần tử của dãy *A* có thứ tự tăng dần theo giá trị.

```
File Edit Format Run Options Window Help
def Merge(A,T,P):
    i = j = k = 0
    #Sao chép các phần tử từ nhỏ đến lớn
    #từ dãy T và P vào dãy A
    while i < len(T) and j < len(P):
        if T[i] < P[j]:
            A[k] = T[i]
            i += 1
        else:
            A[k] = P[j]
            j += 1
        k += 1
    #Kiểm tra xem còn lại phần tử nào trong T và P không
    while i < len(T):
        A[k] = T[i]
        i += 1
        k += 1
    while j < len(P):
        A[k] = P[j]
        j += 1
        k += 1
```

Hình 5. Hàm **`Merge(A,T,P)`** *trộn hai dãy T và P vào dãy A*

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

Bảng 1. Một số bộ dữ liệu thử nghiệm cho thuật toán sắp xếp trộn

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	8 15 3 5 1 27 9 5 11	Dãy sau khi sắp xếp là: 1 3 5 5 9 11 15 27
2	10 9 11 44 7 2 8 11 17 31 7	Dãy sau khi sắp xếp là: 2 7 7 8 9 11 11 17 31 44

Sắp xếp trộn là thuật toán đệ quy điển hình cho mô hình tổng quát của phương pháp chia để trị bao gồm ba bước chính là: chia nhỏ ra thành hai bài toán con; giải từng bài toán con bằng đệ quy; kết hợp kết quả các bài toán con.

3 Mô hình tổng quát của phương pháp chia để trị

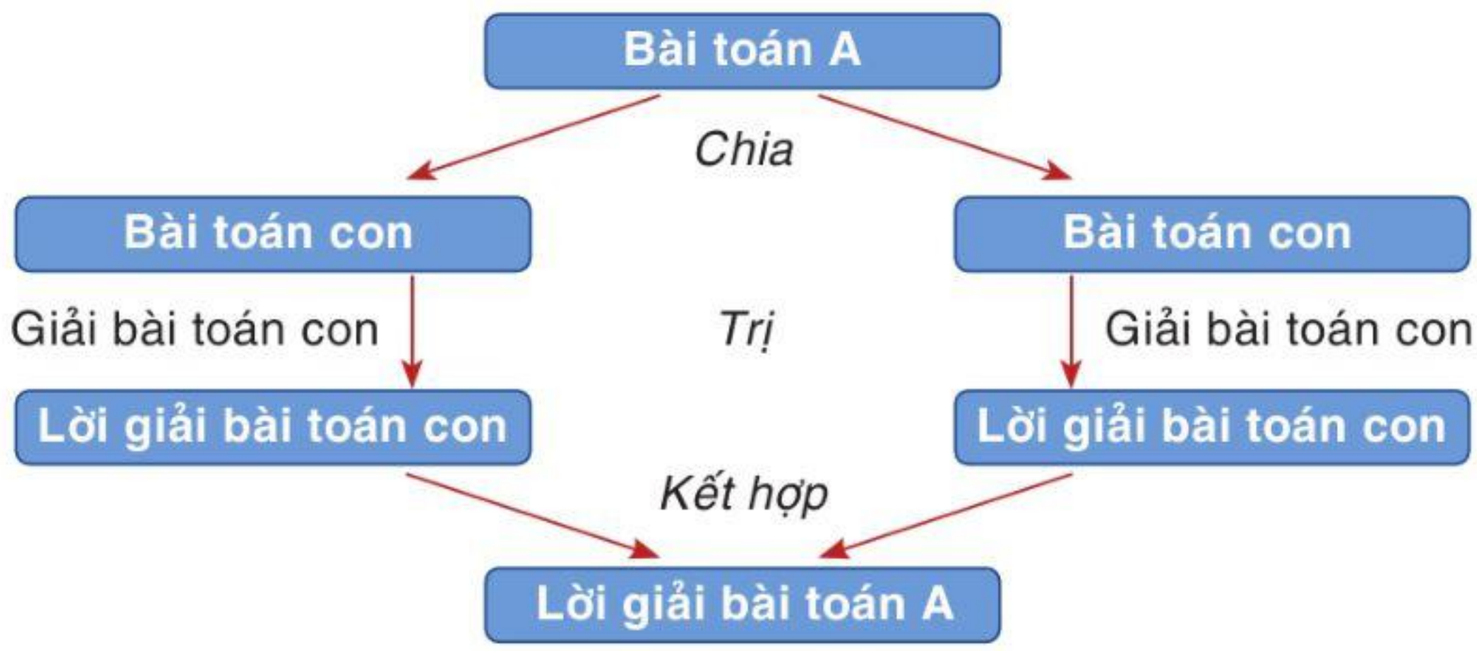
Chia để trị là một mô hình giải bài toán theo hướng làm dễ bài toán đi bằng cách chia thành các phần nhỏ hơn và xử lí từng phần một. Thông thường, các bước cơ bản của một thuật toán chia để trị bao gồm:

1. Chia: Chia bài toán đang giải (bài toán cha) thành một hay nhiều bài toán con giống bài toán cha chỉ khác nhau về kích thước bài toán.

2. Trị: Giải từng bài toán con theo cách giống như giải bài toán cha, mỗi bài toán cần giải sẽ dễ hơn do kích thước bài toán nhỏ hơn. Giải trực tiếp bài toán con khi kích thước bài toán đủ nhỏ, gọi là trường hợp cơ sở.

3. Kết hợp: Kết hợp lời giải các bài toán con lại thành lời giải của bài toán cha và tiếp tục như vậy đến khi thu được lời giải của bài toán ban đầu.

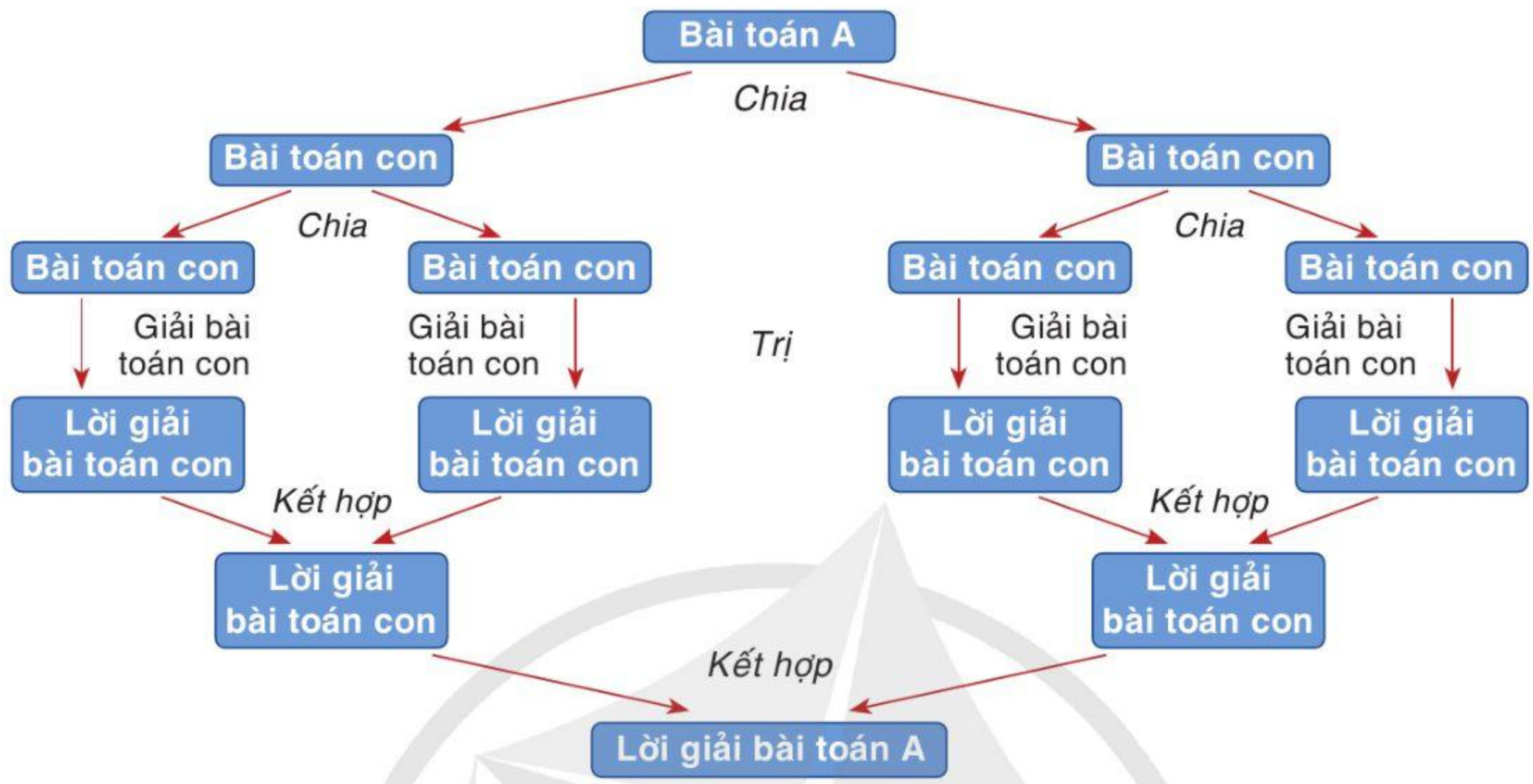
Sơ đồ khối (*Hình 6*) sau đây thể hiện ba bước cơ bản của thuật toán chia để trị trong trường hợp bài toán ban đầu A chia thành hai bài toán con, mỗi bài toán được giải và kết hợp kết quả lại thành kết quả bài toán ban đầu.



Hình 6. Sơ đồ thể hiện ba bước Chia, Trị, Kết hợp của thuật toán chia để trị giải bài toán A khi chia A thành hai bài toán con

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

Nếu bài toán con tiếp tục được chia đệ quy thành hai bài toán con nhỏ hơn, sơ đồ có dạng sau (*Hình 7*).



Hình 7. Sơ đồ thể hiện ba bước Chia, Trị, Kết hợp của thuật toán chia để trị giải bài toán A khi chia bài toán cha thành hai bài toán con tại mỗi bước đệ quy

Nhận xét: Kĩ thuật đệ quy thường được áp dụng trong các bước của thuật toán chia để trị. Mỗi bài toán con được giải bằng cách gọi cùng một thủ tục đệ quy giải bài toán cha với các tham số đầu vào có kích thước nhỏ hơn.

Mô hình chung của phương pháp chia để trị giải một bài toán *A* có thể được thể hiện thông qua đoạn mã giả trong *Hình 8* sau đây.

```
def ChiaDeTriDeQuy(A):
    if (Kích thước của A đủ nhỏ):
        <Giải bài toán A một cách trực tiếp>
    else:
        <Chia bài toán A thành tập k bài toán con (A0,A1,…,Ak-1)>
        for i in range(k):
            ChiaDeTriDeQuy(Ai)
        <Kết hợp các lời giải các bài toán con Ai thành lời giải bài
        toán A>
    return <Lời giải bài toán A>
```

Hình 8. Mô hình phương pháp chia để trị tổng quát

Ưu điểm của kĩ thuật chia để trị là luôn đảm bảo tìm ra nghiệm đúng và thường cho thời gian thực hiện chương trình nhanh. Hạn chế của kĩ thuật này là không phải bài toán nào cũng có thể phân chia thành các bài toán con để có thể thực hiện được kĩ thuật chia để trị.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Em hãy cho biết trong mô tả thuật toán sắp xếp trộn và trong chương trình cài đặt ở trên cần thay đổi thế nào để sắp xếp một dãy theo thứ tự giảm dần của giá trị.

Hội diễn văn nghệ của trường năm nay, lớp Thanh An tham gia biểu diễn khiêu vũ tập thể theo cặp (nam, nữ). Thầy giáo chủ nhiệm chọn ra n bạn nam có chiều cao $A_0, A_1, \ldots, A_{n-1}$ đứng thành một hàng ngang và n bạn nữ có chiều cao $B_0, B_1, \ldots, B_{n-1}$ đứng thành một hàng ngang để ghép thành n cặp (nam, nữ). Để tiện ghép cặp, thầy giáo sắp xếp lại vị trí đứng các bạn nam trong hàng theo thứ tự chiều cao tăng dần và vị trí đứng các bạn nữ trong hàng cũng theo thứ tự chiều cao tăng dần. Sau đó thầy giáo tiến hành ghép cặp bạn nam thấp nhất với bạn nữ thấp nhất, bạn nam thấp thứ hai với bạn nữ thấp thứ hai và cứ như vậy đến bạn nam cao nhất với bạn nữ cao nhất. Em hãy viết chương trình áp dụng thuật toán sắp xếp trộn để giúp thầy giáo thực hiện công việc ghép cặp này.

Chương trình cần nhập vào một số nguyên n, tiếp theo nhập vào n giá trị $A_0, A_1, \ldots, A_{n-1}$ và n giá trị $B_0, B_1, \ldots, B_{n-1}$.

Chương trình cần in ra n cặp số A_i, B_j $(0 \le i, j \le n-1)$ là cách xếp cặp (nam, nữ) theo mong muốn của thầy giáo ở trên.

Trong các câu sau đây, câu nào đúng khi mô tả trình tự các bước cơ bản của phương pháp chia để trị?

a) Chia nhỏ bài toán; Kết hợp kết quả các bài toán con; Giải từng bài toán con bằng đệ quy.

b) Giải bài toán; Chia nhỏ bài toán; Kết hợp các kết quả bài toán.

c) Chia nhỏ bài toán; Giải từng bài toán con bằng đệ quy; Kết hợp kết quả các bài toán con.

Tóm tắt bài học

- Sắp xếp trộn là một thuật toán đệ quy điển hình của phương pháp chia để trị tổng quát bao gồm ba giai đoạn cơ bản: Chia, Trị và Kết hợp.
- Ý tưởng chính của thuật toán sắp xếp trộn là chia đôi dãy cần sắp xếp, đưa bài toán ban đầu về hai bài toán sắp xếp trên dãy có kích thước nhỏ hơn và cuối cùng kết hợp kết quả của hai bài toán lại thành kết quả của bài toán ban đầu.
- Hàm `Merge()` trộn hai dãy đã sắp xếp tăng dần thành một dãy sắp xếp tăng dần.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

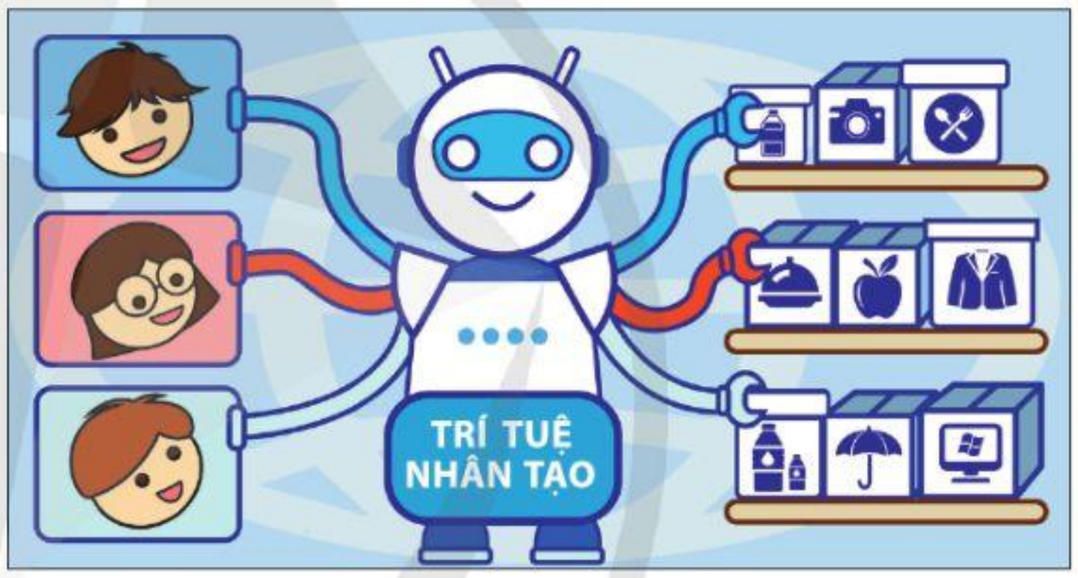
Bài 5

THỰC HÀNH TỔNG HỢP ỨNG DỤNG CHIA ĐỂ TRỊ

Học xong bài này, em sẽ:

- Áp dụng được cách làm thuật toán sắp xếp trộn vào bài toán tính số nghịch thế trong mảng.
- Vận dụng được tư tưởng chia để trị vào giải quyết một số bài toán thực tế.
- Viết được chương trình đơn giản cho một số bài toán áp dụng kĩ thuật đệ quy trong chia để trị.

Các trang web thương mại điện tử ứng dụng trí tuệ nhân tạo thường lưu lại lịch sử thói quen khách hàng khi tương tác với các sản phẩm. Lịch sử tương tác của mỗi khách hàng sẽ được lưu vào một mảng chứa các hành động như: xem, bấm nút "thích" ("like"), mua hàng,... theo thứ tự thời gian. Một phương pháp đơn giản và hiệu quả để gợi ý cho khách hàng là áp dụng bài toán đếm số nghịch thế trong mảng. Chương trình sẽ so sánh mảng của khách hàng đang tương tác trên trang web với mảng của các khách hàng trong lịch sử để đưa ra gợi ý dựa trên số nghịch thế giữa hai mảng (*Hình 1*). Bài toán đếm số nghịch thế trong mảng được phát biểu sau đây.



Hình 1. Minh hoạ hệ thống gợi ý trong thương mại điện tử

Bài toán *Tính số nghịch thế trong mảng*

Cho một mảng A gồm n phần tử $A_0, A_1, \ldots, A_{n-1}$ đôi một khác nhau. Hai phần tử A_i, A_j với $i < j$ được gọi là nghịch thế nếu như $A_i > A_j$. Hãy tính số lượng nghịch thế trong mảng.

Yêu cầu:

Sản phẩm của mỗi nhóm qua bài thực hành này bao gồm:

Thực hành 1: Mô tả chi tiết thuật toán giải quyết bài toán.

Thực hành 2: Viết chương trình giải quyết bài toán.

Thực hành 3: Kết quả thử nghiệm trên các bộ dữ liệu đầu vào mẫu và tự tạo.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Thực hành 1: Áp dụng trực tiếp thuật toán sắp xếp trộn ở trên để mô tả chi tiết phương pháp chia để trị giải bài toán đếm số lượng nghịch thế ở trên.

Hướng dẫn: Khi chia mảng A thành hai mảng con T và P, số lượng nghịch thế từ những cặp phần tử mà một phần tử thuộc mảng T và một phần tử thuộc mảng P sẽ không thay đổi nếu ta sắp xếp các phần tử trong từng mảng T và P tăng dần. Nhận xét này giúp cho chúng ta có thể áp dụng trực tiếp các bước của thuật toán sắp xếp trộn ở trên như sau:

1. *Chia:* Sử dụng thuật toán chia của hàm `Merge_sort(A)`.
2. *Trị:* Gọi đệ quy hàm `Merge_sort(T)` và `Merge_sort(P)` để giải từng bài toán con, đồng thời cập nhật kết quả đếm số lượng nghịch thế từng bài toán con vào một biến đếm.
3. *Kết hợp:* Sử dụng thuật toán trộn của hàm `Merge(A)`, đồng thời cập nhật số lượng nghịch thế từ những cặp phần tử mà một phần tử thuộc mảng T và một phần tử thuộc mảng P, các phần tử trong hai mảng này đều đã được sắp xếp tăng dần.

   Gợi ý: Trong hàm `Merge(A)`, mỗi khi xảy ra điều kiện `T[i]>P[j]`, nghĩa là các phần tử từ `T[i+1]` đến phần tử cuối cùng của mảng `T` đều lớn hơn `P[j]`, số lượng nghịch thế cần được cập nhật thêm là `len(T)-i`, với `len(T)` là số lượng phần tử của mảng `T`.

Em hãy mô tả chi tiết kết quả từng bước trong các hướng dẫn 1, 2, 3 trên cho một trường hợp mảng số cụ thể, ví dụ thực hành trên mảng gồm 7 số có các giá trị lần lượt là 9, 44, 7, 2, 8, 17, 31.

Thực hành 2: Viết chương trình nhập vào một số nguyên dương n và n giá trị $A_0, A_1, \ldots, A_{n-1}$ đôi một khác nhau, đưa ra số lượng nghịch thế của mảng vừa nhập vào.

Hướng dẫn: Sử dụng chương trình thuật toán sắp xếp trộn trong Bài 4 và phần hướng dẫn thuật toán trong Thực hành 1 để hoàn thiện chương trình cho bài toán này.

Kiểm thử chương trình:

Em hãy nhập vào một số ví dụ mảng đầu vào và đưa ra kết quả để kiểm thử chương trình có cho kết quả đúng hay không.

Nếu kết quả kiểm thử trên một số bộ dữ liệu bị sai thì in ra các giá trị trung gian trong chương trình để quan sát sự thay đổi theo từng bước của thuật toán.

Em hãy tạo một mảng đầu vào có kích thước lớn (n khoảng 1 triệu phần tử) và được sắp xếp giảm dần. Từ đó thử chạy chương trình với mảng đầu vào đó.

Bảng 1 là một số ví dụ thử nghiệm, em hãy tự tạo các ví dụ thử nghiệm khác.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bảng 1. Một số bộ dữ liệu thử nghiệm cho Thực hành 2

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	8 15 3 5 1 27 9 7 11	Số lượng nghịch thế của mảng là 12
2	7 9 44 7 2 8 17 31	Số lượng nghịch thế của mảng là 9
3	6 2 8 9 17 31 44	Số lượng nghịch thế của mảng là 0

Giải thích ví dụ:

– Trong bộ dữ liệu thử nghiệm thứ nhất có 12 nghịch thế là các cặp: (15, 3), (15, 5), (15, 1), (15, 9), (15, 7), (15, 11), (3, 1), (5, 1), (27, 9), (27, 7), (27, 11), (9, 7).

– Trong bộ dữ liệu thử nghiệm thứ hai có 9 nghịch thế là các cặp: (9, 7), (9, 2), (9, 8), (44, 7), (44, 2), (44, 8), (44, 17), (44, 31), (7, 2).

– Bộ dữ liệu thử nghiệm thứ ba có các phần tử trong mảng đã được sắp xếp tăng dần nên không có nghịch thế nào.

Thực hành 3: Viết chương trình thực hiện thuật toán đơn giản bằng vòng lặp để đếm số lượng nghịch thế. Tiếp theo, em hãy đếm số bước thực hiện bởi thuật toán này so với thuật toán chia để trị ở trên trong một số ví dụ cụ thể.

Hướng dẫn:

Bước 1. Cài đặt thuật toán đơn giản sử dụng hai vòng lặp lồng nhau duyệt qua tất cả các cặp hai phần tử của mảng dãy số để kiểm tra xem từng cặp hai phần tử có phải là nghịch thế hay không. Đồng thời thêm các biến đếm đặt trong vòng lặp thứ để tính số bước thực hiện của chương trình.

Bước 2. Bổ sung các biến đếm vào vị trí thích hợp trong hàm đệ quy của chương trình trong bài Thực hành 2 để tính số bước thực hiện của chương trình.

Cho một mảng A gồm n phần tử $A_0, A_1, \ldots, A_{n-1}$, hai phần tử bất kì có thể bằng nhau. Hãy tính số lượng những cặp hai phần tử mà không phải là nghịch thế trong mảng.

a) Vận dụng bài Thực hành 1 ở trên để mô tả chi tiết phương pháp chia để trị cho bài toán này.

b) Viết chương trình nhập vào giá trị n và n giá trị $A_0, A_1, \ldots, A_{n-1}$, đưa ra số lượng các cặp không phải là nghịch thế trong mảng A.

c) Tạo các bộ dữ liệu thử nghiệm để kiểm thử chương trình.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

CHUYÊN ĐỀ 3

THỰC HÀNH THIẾT KẾ THUẬT TOÁN THEO KĨ THUẬT DUYỆT

Bài 1

KĨ THUẬT DUYỆT

Học xong bài này, em sẽ:

- Nêu được ý tưởng của kĩ thuật duyệt.
- Tìm hiểu được một số bài toán sử dụng kĩ thuật duyệt.
- Biết được các bước cần thực hiện khi giải bài toán bằng kĩ thuật duyệt.

Để bảo mật thông tin khi trao đổi, một nhóm bạn đã thống nhất mã hoá các số nguyên dương bằng các thanh ngang và chấm tròn, trong đó mỗi thanh ngang có giá trị là 1, mỗi chấm tròn có giá trị là 3. Em hãy cho biết đáp án nào sau đây biểu diễn cho số 8.

A.

B.

C.

D.

1 Ý tưởng kĩ thuật duyệt

Để tìm đáp án cho một bài toán, ta có thể thử tất cả các trường hợp xảy ra rồi chọn phương án đúng. Cách làm như vậy được gọi là *kĩ thuật duyệt*. Con người chỉ sử dụng kĩ thuật duyệt khi giải những bài toán có ít trường hợp. Nhờ tốc độ xử lí nhanh, chính xác nên máy tính có thể giúp con người giải các bài toán có số trường hợp lớn. Kĩ thuật duyệt được sử dụng nhiều để giải quyết các bài toán trong tin học.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

1

Cho biết phương trình $x^3 - 2x^2 + x - 2 = 0$ chỉ có một nghiệm nguyên duy nhất, nghiệm đó là một trong các trường hợp dưới đây. Em hãy cho biết đâu là nghiệm nguyên của phương trình, giải thích cách làm và trình bày ưu nhược điểm của cách làm đó.

A. 1 B. –1 C. 2 D. – 2

Ưu điểm của kĩ thuật duyệt là luôn đảm bảo tìm ra nghiệm đúng, cách làm này đơn giản và chính xác. Tuy nhiên, hạn chế của kĩ thuật này là thời gian thực thi lâu nếu số trường hợp phải thử lớn.

2 Bài toán *Chọn mua đồ dùng học tập*

Đầu năm học mới, Hồng cùng mẹ đã đặt mua trực tuyến các đồ dùng học tập cần thiết. Để thưởng cho Hồng vì thành tích học tập năm trước, mẹ cho phép Hồng lựa chọn mua thêm một đồ dùng nữa có giá không vượt quá T (đồng). Trên trang web hiện ra n đồ dùng, các đồ dùng được đánh số từ 0 đến $n - 1$ với mức giá tương ứng là $p_0, p_1, \ldots, p_{n-1}$ và có mức độ yêu thích của Hồng tương ứng là $s_0, s_1, \ldots, s_{n-1}$. Em hãy lập trình nhập vào hai số nguyên n, T cùng với hai dãy $p_0, p_1, \ldots, p_{n-1}$ và $s_0, s_1, \ldots, s_{n-1}$, đưa ra một đồ dùng có giá không vượt quá T (đồng) mà Hồng yêu thích nhất.

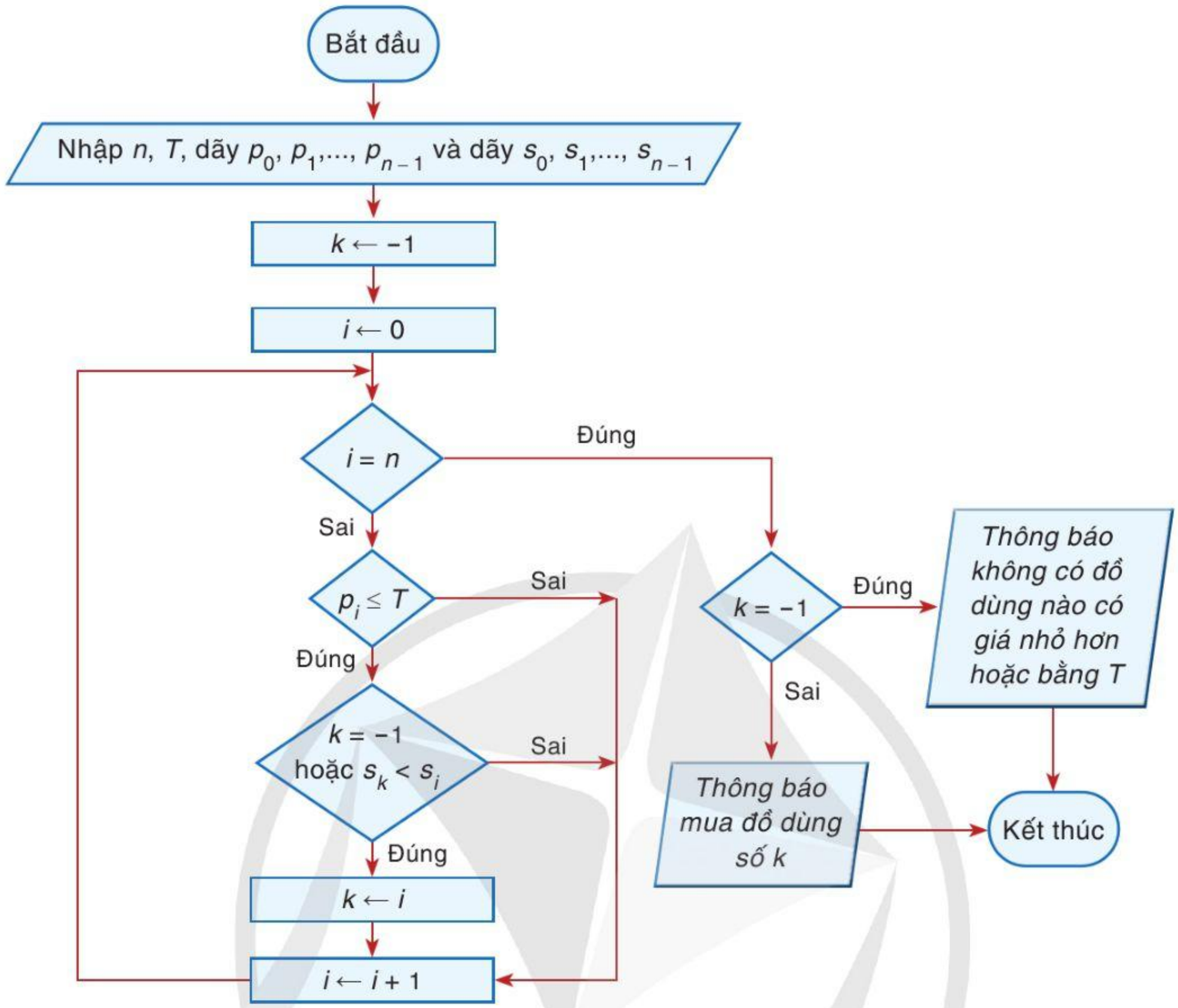
Ví dụ, một số đồ dùng học tập có giá và mức độ yêu thích tương ứng (*Hình 1*).

Hình 1. Ví dụ cho bài toán Chọn mua đồ dùng học tập

Để tìm được đồ dùng yêu thích nhất có giá không vượt quá T (đồng), Hồng sẽ lần lượt xét từng đồ vật. Với mỗi đồ vật, Hồng xem giá của đồ vật đó, nếu đồ vật đó có giá lớn hơn T (đồng), Hồng sẽ chuyển qua đồ vật tiếp theo; ngược lại, nếu đồ vật có giá nhỏ hơn hoặc bằng T (đồng), Hồng sẽ xem đồ dùng đó có được yêu thích hơn đồ dùng trước đó Hồng đã lựa chọn không, nếu đồ dùng đó được yêu thích hơn thì Hồng lựa chọn đồ dùng này và trả lại đồ dùng trước đó đã chọn.

Cứ tiếp tục như vậy cho đến khi xét hết n đồ dùng. Thuật toán được mô tả bằng sơ đồ khối trong *Hình 2*.

Đọc bản mới nhất trên hoc10.vn Bản mẫu



Hình 2. Sơ đồ khối giải bài toán Chọn mua đồ dùng học tập

2

Em hãy tìm hiểu chương trình viết trên ngôn ngữ lập trình Python (*Hình 3*) giải bài toán trên, soạn thảo và chạy với các bộ dữ liệu thử nghiệm trong *Bảng 1*.

File Edit Format Run Options Window Help

```python
n, T = map(int, input().split())
p = list(map(int, input().split()))
s = list(map(int, input().split()))
k = -1
for i in range(n):
  if p[i] <= T:
    if (k == -1) or (s[k] < s[i]):
      k = i
if (k == -1):
  print("Không có đồ dùng học tập nào có giá nhỏ hơn hoặc bằng", T)
else:
  print("Chọn mua đồ dùng số", k)
```

Hình 3. Chương trình cho thuật toán ở Hình 2

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

Bảng 1. Một số bộ dữ liệu thử nghiệm cho Hoạt động 2

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	5 100 60 110 100 70 90 5 9 7 7 8	Chọn mua đồ dùng số 4
2	5 100 60 110 100 70 90 5 9 9 7 8	Chọn mua đồ dùng số 2
3	5 50 60 110 100 70 90 5 9 7 7 8	Không có đồ dùng học tập nào có giá nhỏ hơn hoặc bằng 50

3 Các bước thực hiện kĩ thuật duyệt

Xét bài toán *Chọn mua hai đồ dùng học tập* tương tự bài toán *Chọn mua đồ dùng học tập* nhưng thay vì chỉ mua một đồ dùng thì Hồng cần mua đúng hai đồ dùng với tổng giá không vượt quá T (đồng) và tổng độ yêu thích của hai đồ dùng đó là lớn nhất.

Trong bài toán này, lời giải bài toán không chỉ có một chỉ số mà phải cần đến hai chỉ số để mô tả hai đồ dùng được chọn, ta có thể biểu diễn bằng cặp chỉ số (i, j) với $0 \leq i < j \leq n - 1$, trong đó i là chỉ số đồ dùng thứ nhất, j là chỉ số đồ dùng thứ hai. Để tìm hai đồ dùng có tổng giá không vượt quá T (đồng) và tổng độ yêu thích của hai đồ dùng đó là lớn nhất ta có thể duyệt toàn bộ các cặp chỉ số (i, j) với $0 \leq i < j \leq n - 1$, có $\frac{n(n-1)}{2}$ cặp như vậy. Với mỗi cặp, tiến hành tính tổng giá và kiểm tra $p_i + p_j$ không vượt quá T (đồng) mà tổng độ yêu thích $s_i + s_j$ tốt hơn lựa chọn đã chọn thì cập nhật lại lựa chọn.

Như vậy, để giải bài toán bằng kĩ thuật duyệt cần thực hiện các bước sau:

1. Mô tả lời giải của bài toán: Biểu diễn từng thành phần của lời giải bằng các biến và chỉ rõ miền giá trị của chúng.

2. Kiểm tra và chọn nghiệm: Liệt kê tất cả các khả năng theo cách đã mô tả, với mỗi khả năng kiểm tra điều kiện để chọn nghiệm.

Câu 1. Chương trình trong *Hình 4* giải bài toán *Chọn mua hai đồ dùng học tập*. Em hãy cho biết dấu ? cần được thay bằng gì để chương trình chạy đúng. Em hãy soạn thảo, hoàn thiện chương trình và chạy với các bộ dữ liệu thử nghiệm trong *Bảng 2*.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

```
File Edit Format Run Options Window Help
n, T = map(int, input().split())
p = list(map(int, input().split()))
s = list(map(int, input().split()))
k1 = -1
k2 = -1
for i in range(n):
  for j in range(i+1, n):
    print("xét hai đồ dùng :", i, j)
    if p[i] + p[?] <= T:
      if (k1 == -1) or (s[k1] + s[k2] < s[i] + s[j]):
        k1 = i
        k2 = j
if (k1 == -1):
  print("Không có hai đồ dùng học tập nào có tổng giá nhỏ hơn hoặc bằng", T)
else:
  print("Chọn mua hai đồ dùng số", k1, "và", k2)
```

Hình 4. Chương trình giải bài toán Chọn mua hai đồ dùng học tập

Bảng 2. Một số bộ dữ liệu thử nghiệm cho Câu 1 (Luyện tập)

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	5 200 60 110 100 70 90 5 9 7 7 8	Chọn mua đồ dùng số 1 và 4
2	5 160 60 110 100 70 90 5 9 9 7 8	Chọn mua đồ dùng số 3 và 4
3	5 120 60 110 100 70 90 5 9 7 7 8	Không có hai đồ dùng học tập nào có tổng giá nhỏ hơn hoặc bằng 120

Câu 2. Xét phương trình $ax^5 + bx + c = 0$ với a, b, c là các hằng số nguyên khác 0 có giá trị tuyệt đối không vượt quá 10^6. Theo lược đồ Hoocne, nghiệm nguyên của đa thức sẽ là ước của c. Em hãy lập trình nhập vào ba số nguyên a, b, c. Sau đó, lập trình tìm tất cả các nghiệm nguyên của phương trình và chạy với các bộ dữ liệu thử nghiệm trong *Bảng 3*.

Bảng 3. Một số bộ dữ liệu thử nghiệm cho Câu 2 (Luyện tập)

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	1 1 2	Phương trình có các nghiệm nguyên là: −1
2	1 1 −2	Phương trình có các nghiệm nguyên là: 1
3	1 −15 −2	Phương trình có các nghiệm nguyên là: 2

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

Một lớp học nhảy có n bạn nam và n bạn nữ. Bạn nam thứ i $(0 \le i \le n-1)$ có chiều cao là b_i (cm), bạn nữ thứ j $(0 \le j \le n-1)$ có chiều cao là g_j (cm). Trong một buổi học, thầy giáo muốn chọn ra một đôi nhảy gồm một bạn nam và một bạn nữ để trình diễn mà bạn nam cao hơn bạn nữ và chênh lệch độ cao của hai bạn là nhỏ nhất. Em hãy lập trình nhập vào hai dãy số $b_0, b_1, \ldots, b_{n-1}$ và $g_0, g_1, \ldots, g_{n-1}$, đưa ra hai số tương ứng là chiều cao của bạn nam và bạn nữ được chọn. Sau đó, chạy chương trình với các bộ dữ liệu trong *Bảng 4*.

Bảng 4. Một số bộ dữ liệu thử nghiệm cho Vận dụng

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	180 173 175 167 173 169	175 173
2	180 173 175 170 167 173 169 170	170 169
3	170 175 180 185 165 168 171 173	175 173

Trong những câu sau đây, câu nào đúng khi nói về kĩ thuật duyệt?

a) Kĩ thuật duyệt chỉ cần thử một số trường hợp trong các trường hợp có thể xảy ra là xác định được nghiệm của bài toán.

b) Kĩ thuật duyệt sẽ xét tất cả các trường hợp có thể xảy ra để xác định được nghiệm của bài toán.

c) Kĩ thuật duyệt không đảm bảo tìm được nghiệm đúng.

d) Kĩ thuật duyệt chạy rất nhanh trong mọi trường hợp.

Tóm tắt bài học

- Kĩ thuật duyệt được dùng để giải quyết nhiều bài toán. Ưu điểm của kĩ thuật duyệt là luôn đảm bảo tìm ra nghiệm đúng, nhưng thời gian thực thi lâu nếu số trường hợp phải thử lớn.
- Hai bước cần thực hiện để giải bài toán bằng kĩ thuật duyệt: mô tả lời giải của bài toán; kiểm tra và chọn nghiệm.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bài 2

KĨ THUẬT QUAY LUI

Học xong bài này, em sẽ:

- Tìm hiểu được chương trình liệt kê dãy bit độ dài *n* bằng kĩ thuật đệ quy.
- Nêu được ý tưởng của kĩ thuật quay lui.
- Tìm hiểu được lời giải một số bài toán sử dụng kĩ thuật quay lui.

Trong bài học trước, các em đã tìm hiểu bài toán *Chọn mua đồ dùng học tập* với các tình huống mua một đồ dùng hoặc hai đồ dùng. Nếu bài toán không cố định số lượng đồ dùng cần mua mà có thể mua một số đồ dùng với tổng giá không vượt quá *T* (đồng) và tổng mức độ yêu thích của các đồ dùng đó là lớn nhất, em hãy trình bày ý tưởng giải quyết bài toán.

1 Bài toán *Mua đồ tổng quát*

Bài toán: Cho n đồ dùng với mức giá tương ứng là $p_0, p_1, \ldots, p_{n-1}$ (đồng) và có mức độ yêu thích của Hồng tương ứng là $s_0, s_1, \ldots, s_{n-1}$. Em hãy giúp Hồng chọn mua một số đồ dùng với tổng giá không vượt quá T (đồng) mà tổng mức độ yêu thích là lớn nhất.

1

Cho 5 đồ dùng với giá và mức độ yêu thích tương ứng như trong *Bảng 1*. Nếu $T = 20$ (nghìn đồng) thì Hồng cần chọn mua những đồ dùng nào để tổng mức độ yêu thích là lớn nhất?

Bảng 1. Thông tin về các đồ dùng

Đồ dùng	0	1	2	3	4
Mức giá (nghìn đồng)	10	5	18	9	5
Mức độ yêu thích	7	2	12	6	3

Lời giải của bài toán này có thể biểu diễn bằng một dãy bit độ dài n (n là số lượng đồ vật), trong đó bit thứ i ($0 \le i \le n-1$) bằng 1 hoặc 0 tương ứng là vật thứ i được chọn hoặc không chọn.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Ví dụ, dãy bit (1, 0, 0, 1, 0) tương ứng với cách chọn đồ dùng số 0 và 3 với tổng giá là 10 + 9 = 19 (nghìn đồng) và mức độ yêu thích là 7 + 6 = 13; dãy bit (1, 1, 0, 0, 1) tương ứng với cách chọn đồ dùng số 0, 1 và 4 có tổng giá là 10 + 5 + 5 = 20 (nghìn đồng) và mức độ yêu thích là 7 + 2 + 3 = 12.

Để giải quyết bài toán *Mua đồ tổng quát* bằng kĩ thuật duyệt ta có thể xét toàn bộ dãy bit độ dài *n*, với mỗi dãy bit tương ứng với một phương án mua, ta tiến hành tính tổng giá để kiểm tra ràng buộc không vượt quá *T* (đồng) và tính tổng mức độ yêu thích để chọn phương án tối ưu.

2 Liệt kê dãy bit độ dài *n* bằng kĩ thuật đệ quy

2

Em hãy tìm hiểu chương trình liệt kê dãy bit độ dài *n* bằng kĩ thuật đệ quy trong *Hình 1* và chạy thử nghiệm chương trình. Cho biết số lượng dãy bit nhị phân độ dài 3, 5, 10 tương ứng là bao nhiêu.

File Edit Format Run Options Window Help

```
1  def backtrack(i):
2      for v in range(2):
3          x.append (v)
4          if i == n - 1:
5              print(x)
6          else:
7              backtrack(i + 1)
8          x.pop()
9
10 n = int(input())
11 x = []
12 backtrack(0)
```

```
3
[0, 0, 0]
[0, 0, 1]
[0, 1, 0]
[0, 1, 1]
[1, 0, 0]
[1, 0, 1]
[1, 1, 0]
[1, 1, 1]
```

Hình 1. Chương trình liệt kê dãy bit bằng kĩ thuật đệ quy và màn hình kết quả chạy chương trình với n = 3

Dãy bit độ dài *n* có dạng $X = (x_0, x_1, ..., x_{n-1})$, trong đó x_i bằng 0 hoặc 1 ($0 \le i \le n - 1$) có thể mô tả theo cách đệ quy như sau:

– Nếu $n > 0$ thì phần tử đầu tiên của dãy bằng 0 hoặc 1 và $n - 1$ phần tử sau là dãy bit độ dài $n - 1$.

– Ngược lại, nếu $n = 0$ thì dãy bit độ dài *n* là dãy rỗng.

Việc xây dựng các dãy nhị phân theo thuật toán đệ quy như sau:

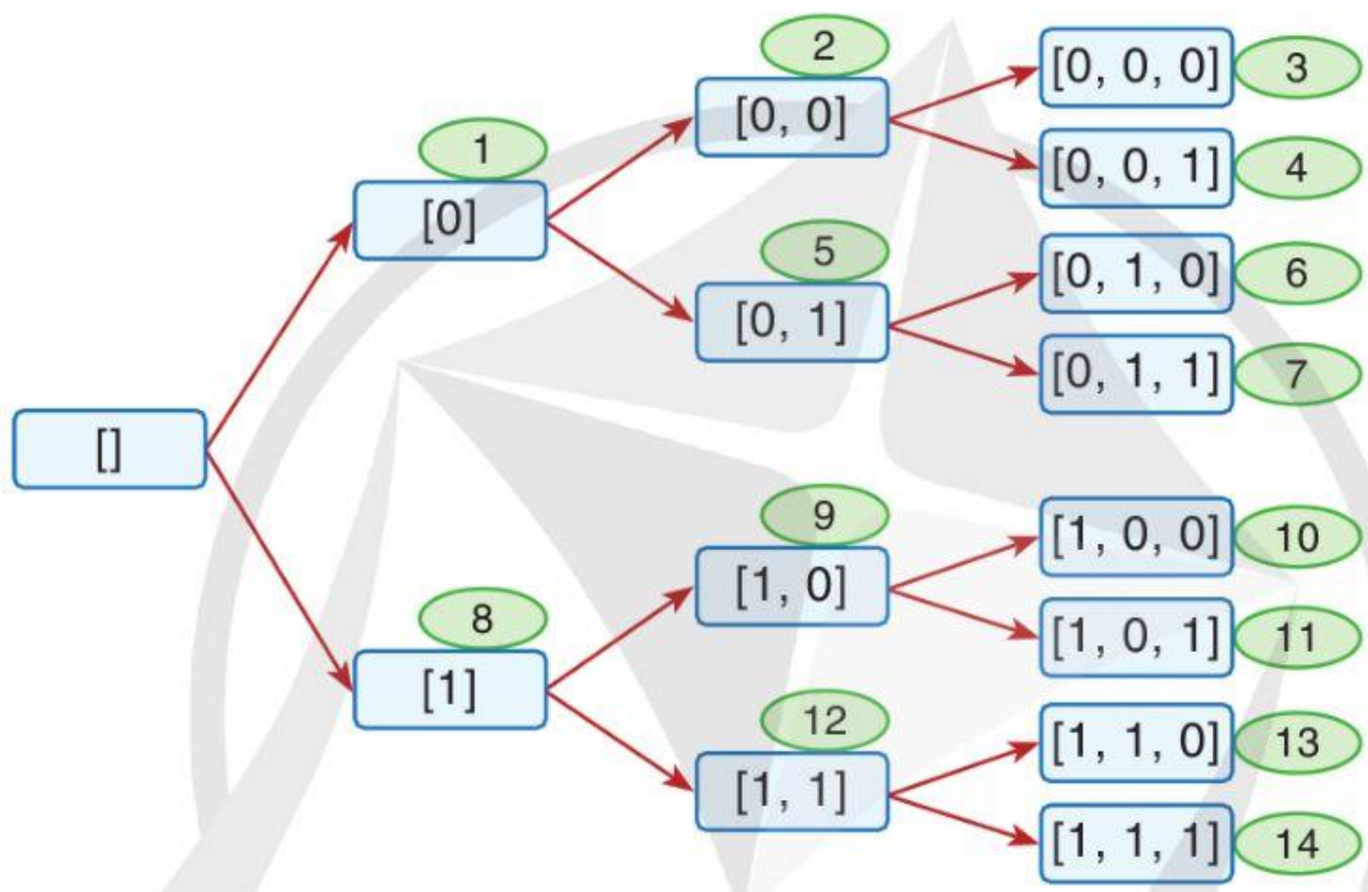
1. Bắt đầu từ *X* rỗng, lệnh **x** = [] và gọi thủ tục đệ quy **backtrack(0)** để xây dựng bắt đầu từ phần tử 0.
2. Thành phần *i* ($0 \le i \le n - 1$) sẽ lần lượt nhận giá trị 0 và 1 bằng lệnh **for v in range(2):**. Với mỗi giá trị của *v*, thành phần *i* được ghi nhận vào x_i của *X* bằng

Đọc bản mới nhất trên hoc10.vn Bản mẫu

lệnh **x.append(v)**, lệnh này đẩy *v* vào cuối *X*. Sau đó tiếp tục gọi đệ quy để xây dựng các thành phần còn lại (từ thành phần x_{i+1} đến thành phần x_{n-1}).

3. Để xét được khả năng tiếp theo, hành động quay lui được thực hiện bằng cách loại bỏ ghi nhận thành phần cuối cùng của *X* bằng lệnh **x.pop()**. Việc quay lui cũng được diễn ra khi đang xây dựng thành phần x_i mà x_i đã lần lượt nhận cả hai giá trị 0 và 1, khi đó thành phần x_i sẽ bị loại khỏi *X* và lùi về để xét khả năng tiếp theo cho thành phần x_{i-1}.

Hình 2 dưới đây mô tả quá trình xây dựng các dãy nhị phân với độ dài $n = 3$. Dãy trong hình chữ nhật là dãy bit trong quá trình xây dựng, số trong hình ô van là thứ tự gọi đệ quy trong quá trình xây dựng.



Hình 2. Quá trình gọi đệ quy để xây dựng dãy bit độ dài 3

3 Kĩ thuật quay lui

Giả sử lời giải của bài toán có dạng $X = (x_0, x_1, ..., x_{n-1})$ với $x_i \in C_i$ (trong đó C_i là tập các giá trị có thể của x_i), để xét tất cả các khả năng của lời giải, ta xây dựng lời giải dần từng bước:

1. Bắt đầu từ lời giải rỗng [].

2. Giả sử, hiện tại đang xây dựng được *i* thành phần $(x_0, x_1, ..., x_{i-1})$, để xây dựng thành phần x_i, cần xét từng khả năng trong C_i.

– Nếu xây dựng xong lời giải thì lời giải hiện tại sẽ được kiểm tra đánh giá và chọn nghiệm. Nếu chưa xây dựng xong lời giải thì xây dựng tiếp thành phần x_{i+1}.

– Nếu đã xét xong các khả năng cho thành phần x_i thì quay lui xét khả năng tiếp theo của thành phần x_{i-1}.

Quá trình sẽ dừng lại khi tất cả các khả năng lựa chọn của các thành phần của lời giải đều đã được xét.

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

Khi cài đặt kĩ thuật quay lui, người ta sử dụng kĩ thuật đệ quy để xây dựng tất cả các khả năng của lời giải, với mỗi khả năng kiểm tra đánh giá để chọn nghiệm. Mô hình kĩ thuật quay lui sử dụng kĩ thuật đệ quy được mô tả trong *Hình 3*.

```
def backtrack(i): #Xây dựng thành phần số i
    for xi in <các_khả_năng các giá trị có thể của xi>:
        <ghi nhận thành phần số i>
        if (<xây dựng xong>):
            <Kiểm tra, đánh giá và chọn nghiệm>
        else:
            backtrack(i+1)
        <Bỏ ghi nhận thành phần số i>
```

Hình 3. Mô hình kĩ thuật quay lui

Em hãy tìm hiểu, soạn thảo chương trình giải bài toán *Mua đồ tổng quát* trong *Hình 4* bằng kĩ thuật quay lui và chạy thử nghiệm với các bộ dữ liệu trong *Bảng 2*.

File Edit Format Run Options Window Help

```
1  def updateSolution():
2      global bestS, bestSol
3      sumS = 0
4      sumP = 0
5      for i in range(n):
6          if x[i] == 1:
7              sumS = sumS + s[i]
8              sumP = sumP + p[i]
9      if (sumP <= T) and (sumS > bestS):
10         bestS = sumS
11         bestSol = list(x)
12 def backtrack(i):
13     for v in range(2):
14         x.append(v)
15         if i == n - 1:
16             updateSolution()
17         else:
18             backtrack(i + 1)
19         x.pop()
20
21 n, T = map(int, input().split())
22 p = list(map(int, input().split()))
23 s = list(map(int, input().split()))
24 x = []
25 bestS = -1
26 bestSol = []
27 backtrack(0)
28 print(bestS, bestSol)
```

Hình 4. Chương trình giải bài toán Mua đồ tổng quát bằng kĩ thuật quay lui

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bảng 2. Một số bộ dữ liệu thử nghiệm cho bài toán Mua đồ tổng quát

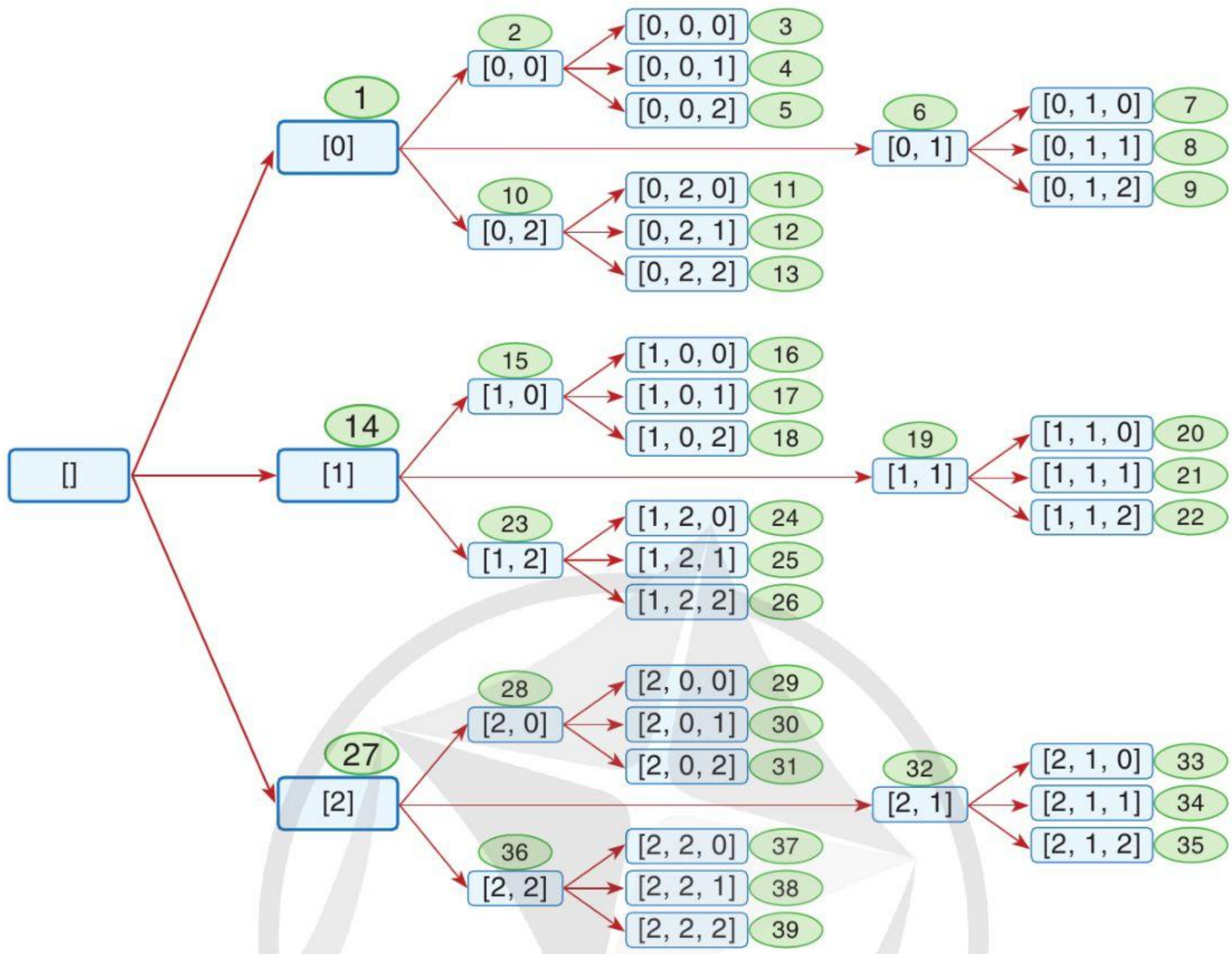
BỘ DỮ LIỆU	DỮ LIỆU VÀO	KẾT QUẢ RA	GIẢI THÍCH
1	5 18 10 5 18 9 5 7 2 12 6 3	12 [0, 0, 1, 0, 0]	Chọn đồ dùng số 2, tổng giá là 18 với tổng mức độ yêu thích là 12.
2	5 20 10 5 18 9 5 7 2 12 6 3	13 [1, 0, 0, 1, 0]	Chọn đồ dùng số 0 và số 3, tổng giá là 19 với tổng mức độ yêu thích là 13.
3	5 33 10 5 18 9 5 7 2 12 6 3	22 [1, 0, 1, 0, 1]	Chọn đồ dùng số 0, số 2 và số 4, tổng giá là 33 với tổng mức độ yêu thích là 22.

Hồng có n tệp dữ liệu được đánh số từ 0 đến $n - 1$ và có kích thước tương ứng là $s_0, s_1, \ldots, s_{n-1}$ (Mb). Hồng muốn tìm cách lưu trữ được nhiều tệp dữ liệu nhất bằng hai đĩa ổ cứng, mỗi ổ có dung lượng D (Mb). Em hãy lập trình giúp Hồng giải quyết bài toán trên, chương trình sẽ nhập vào số nguyên D và dãy số $s_0, s_1, \ldots, s_{n-1}$, sau đó đưa ra phương án lưu trữ là một dãy số $x_0, x_1, \ldots, x_{n-1}$, trong đó x_i $(0 \le i \le n - 1)$ nhận một trong ba giá trị 0 (không được lưu trữ), 1 (lưu trên ổ cứng thứ nhất) hoặc 2 (lưu trên ổ cứng thứ hai). Xem *Hình 5* mô tả quá trình xây dựng các dãy X. Chạy thử nghiệm với các bộ dữ liệu trong *Bảng 3*.

Bảng 3. Một số bộ dữ liệu thử nghiệm cho bài toán Mua đồ tổng quát

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	50000 10000 20000 30000 40000	1 2 2 1
2	50000 10000 4000 20000 30000	1 1 2 2
3	80000 10000 20000 30000 40000 50000	1 1 2 2 1
4	100000 40000 50000 60000 70000 50000	1 2 1 0 2

Đọc bản mới nhất trên hoc10.vn
Bản mẫu



Hình 5. Quá trình xây dựng các khả năng

Trong những câu sau đây, câu nào đúng khi nói về kĩ thuật quay lui?

a) Kĩ thuật quay lui không thể liệt kê tất cả các trường hợp có thể xảy ra để tìm được nghiệm của bài toán.

b) Khi cài đặt kĩ thuật quay lui, bắt buộc phải sử dụng kĩ thuật đệ quy.

c) Kĩ thuật quay lui là một kĩ thuật theo ý tưởng của kĩ thuật duyệt.

Tóm tắt bài học

- Kĩ thuật quay lui xây dựng tất cả các khả năng của lời giải bằng cách mở rộng từng thành phần và quay lui, bắt đầu từ lời giải rỗng. Với cách làm này, kĩ thuật quay lui có thể xét tất cả các khả năng của lời giải và kiểm tra, đánh giá để chọn nghiệm của bài toán theo ý tưởng của kĩ thuật duyệt.
- Khi cài đặt kĩ thuật quay lui, người ta thường sử dụng kĩ thuật đệ quy.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bài 3

THỰC HÀNH KĨ THUẬT QUAY LUI

Học xong bài này, em sẽ:

- Tìm hiểu được chương trình liệt kê các hoán vị của *n* phần tử bằng kĩ thuật đệ quy.
- Tìm hiểu được một số bài toán sử dụng kĩ thuật quay lui.
- Nhận ra được mối liên quan giữa thiết kế thuật toán theo kĩ thuật quay lui và kĩ thuật đệ quy.

Bài toán 1. Trả tiền

Khi đi mua đồ giúp mẹ, Hồng phải trả tổng số tiền là s (nghìn đồng), hiện tại Hồng có n tờ tiền được đánh số từ 0 đến $n - 1$ với mệnh giá tương ứng $t_0, t_1, \ldots, t_{n-1}$ (nghìn đồng). Hồng muốn liệt kê tất cả các cách chọn các tờ tiền mà tổng đúng bằng s (nghìn đồng).

Ví dụ: Với $s = 130$ (nghìn đồng) và có 5 tờ tiền với mệnh giá tương ứng là 10, 20, 50, 50, 100 (nghìn đồng) thì Hồng có hai cách trả. Ví dụ, một cách trả gồm 3 tờ: tờ số 0, tờ số 1 và tờ số 4 có mệnh giá tương ứng là 10, 20, 100 (nghìn đồng), tổng là 10 + 20 + 100 = 130 (nghìn đồng).

DỮ LIỆU VÀO	KẾT QUẢ RA
5 130	0 1 4
10 20 50 50 100	0 1 2 3

Bài toán 2. Liệt kê hoán vị của *n* phần tử bằng kĩ thuật đệ quy

Một hoán vị của 0, 1,..., $n - 1$ là một dãy $(x_0, x_1, \ldots, x_{n-1})$ mà $0 \le x_i \le n - 1$ và các x_i đôi một khác nhau. Ví dụ, $n = 3$ có 6 hoán vị của 0, 1, 2 là các hoán vị: (0, 1, 2), (0, 2, 1), (1, 0, 2), (1, 2, 0), (2, 0, 1), (2, 1, 0). Trong nhiều bài toán, chúng ta cần liệt kê hết tất cả các hoán vị của n phần tử. Em hãy tìm hiểu chương trình liệt kê các hoán vị của n phần tử bằng kĩ thuật đệ quy *Hình 1* và chạy thử nghiệm chương trình.

File Edit Format Run Options Window Help

```
1  def permutation(i):
2    for v in range(n):
3      if not(v in x):
4        x.append(v)
5        if i == n - 1:
6          print(x)
7        else:
8          permutation(i + 1)
9        x.pop()
10
11 n = int(input())
12 x = []
13 permutation(0)
```

```
3
[0, 1, 2]
[0, 2, 1]
[1, 0, 2]
[1, 2, 0]
[2, 0, 1]
[2, 1, 0]
```

Hình 1. Chương trình liệt kê hoán vị của n phần tử bằng kĩ thuật đệ quy và màn hình kết quả chạy chương trình với n = 3

Đọc bản mới nhất trên hoc10.vn

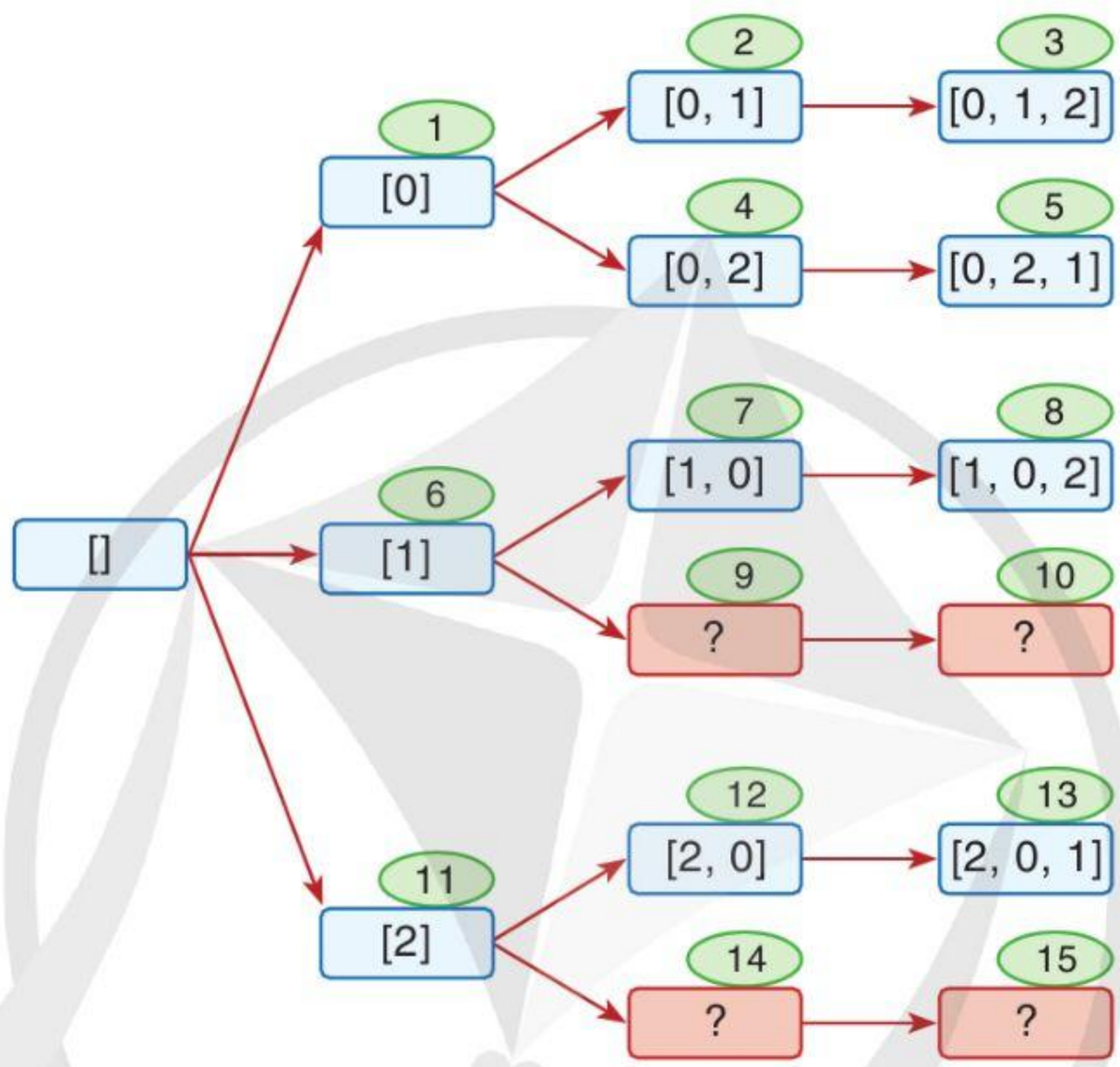
Bản mẫu

a) Cho biết số lượng hoán vị của 3, 4 hoặc 5 phần tử là bao nhiêu.

b) Em hãy so sánh chương trình liệt kê các dãy bit độ dài *n* với chương trình liệt kê các hoán vị của *n* phần tử.

c) Cho biết ý nghĩa của lệnh `x.append(v)` và `x.pop()` trong chương trình.

d) *Hình 2* mô tả quá trình gọi đệ quy để xây dựng các hoán vị 3 phần tử. Dãy ở hình chữ nhật là dãy hoán vị trong quá trình xây dựng, số trong hình ô van là thứ tự xây dựng. Em hãy cho biết các dãy trong hình chữ nhật chứa dấu ?.



Hình 2. Quá trình gọi đệ quy để xây dựng các hoán vị 3 phần tử

Bài toán *Hoán vị các chữ cái*

Em hãy lập trình, nhập vào một từ gồm các chữ cái khác nhau, liệt kê ra tất cả các hoán vị của các chữ cái đó. Chạy thử nghiệm với các bộ dữ liệu ở *Bảng 1*.

Bảng 1. Một số bộ dữ liệu thử nghiệm

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	AB	AB BA
2	TIN	TIN TNI ITN INT NTI NIT

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bài 4

THỰC HÀNH TỔNG HỢP KĨ THUẬT DUYỆT

Học xong bài này, em sẽ:

Viết được một số chương trình đơn giản sử dụng kĩ thuật duyệt.

Bài toán 1. Chọn quả

Có n quả cam, n quả táo và n quả lê ($n \leq 1\ 000$). Em hãy lập chương trình nhập vào số nguyên dương n, sau đó đưa ra các cách lấy n quả, hai cách được gọi là khác nhau nếu tồn tại một loại quả trong cách này có số lượng khác với trong cách kia. Chạy thử nghiệm với các dữ liệu ở *Bảng 1*.

Bảng 1. Một số bộ dữ liệu thử nghiệm cho Bài toán 1

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	1	Có 3 cách
2	100	Có 5151 cách
3	1000	Có 501501 cách

Gợi ý: Gọi x là số quả cam, y là số quả táo và z là số quả lê được lấy ra. Khi đó, số cách chọn là số nghiệm nguyên không âm của phương trình: $x + y + z = n$.

Bài toán 2. Chọn học sinh

Một nhóm học sinh gồm n ($n \leq 10$) bạn, cần chọn k ($k \leq n$) bạn tham gia một trò chơi. Em hãy lập chương trình nhập vào hai số nguyên dương n, k và danh sách gồm n tên học sinh, sau đó liệt kê tất cả các cách chọn k học sinh trong n học sinh. Chạy thử nghiệm với bộ dữ liệu ở *Bảng 2*.

Bảng 2. Một số bộ dữ liệu thử nghiệm cho Bài toán 2

DỮ LIỆU VÀO	KẾT QUẢ RA
3 2 Minh Quang Giang	1. Minh Quang 2. Minh Giang 3. Quang Giang

Gợi ý: Sử dụng kĩ thuật quay lui để xây dựng tất cả các dãy bit độ dài n, với mỗi dãy bit tương ứng với một cách chọn, kiểm tra số lượng bit 1 bằng đúng k hay không?

Một lớp học nhảy có n ($n \leq 10$) bạn nam và n bạn nữ. Bạn nam thứ i ($0 \leq i \leq n - 1$) có chiều cao là b_i (cm), bạn nữ thứ j ($0 \leq j \leq n - 1$) có chiều cao là g_j (cm).

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

Trong một buổi học, thầy giáo muốn ghép các bạn nam với các bạn nữ tạo thành n đôi nhảy để trình diễn, mỗi đôi gồm một bạn nam và một bạn nữ mà chênh lệch chiều cao của hai bạn không vượt quá d (cm). Em hãy lập chương trình đếm số cách ghép thoả mãn, chương trình nhập vào số nguyên d cùng với hai dãy số $b_0, b_1, \ldots, b_{n-1}$ và $g_0, g_1, \ldots, g_{n-1}$, sau đó đưa ra số cách ghép thoả mãn. Chạy thử nghiệm với các bộ dữ liệu ở *Bảng 3*.

Gợi ý: Sử dụng kĩ thuật quay lui để xây dựng tất cả các hoán vị của n phần tử, với mỗi hoán vị tương ứng với một cách ghép, kiểm tra và đếm.

Bảng 3. Một số bộ dữ liệu thử nghiệm cho bài Vận dụng

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	5 170 173 175 167 169 173	1
2	10 170 173 175 167 169 173	6

BÀI TÌM HIỂU THÊM

CÂN ĐĨA

Cho một cân hai đĩa và n ($n \leq 8$) quả cân có khối lượng đôi một khác nhau w_0, $w_1, \ldots, w_{n-1}$ (kg). Tiến hành đặt lần lượt từng quả cân lên một trong hai đĩa của cân và đảm bảo rằng tổng khối lượng bên trái luôn nhỏ hơn hoặc bằng tổng khối lượng bên phải. Em hãy lập trình nhập vào một dãy số nguyên dương $w_0, w_1, \ldots, w_{n-1}$ (kg) là khối lượng của n quả cân, sau đó đưa ra số cách xếp n quả cân thoả mãn. Hai cách được gọi là khác nhau nếu thứ tự xếp các quả cân khác nhau hoặc tồn tại một quả cân nằm ở đĩa khác nhau. Chạy thử nghiệm với các bộ dữ liệu ở *Bảng 4*.

Gợi ý: Sử dụng kĩ thuật quay lui để xây dựng tất cả các hoán vị của n phần tử, với mỗi hoán vị tương ứng với một thứ tự đặt các quả cân, tiếp tục sử dụng kĩ thuật quay lui xây dựng tất cả các dãy bit độ dài n tương ứng với một cách đặt lên đĩa, tiến hành kiểm tra và đếm.

Bảng 4. Một số bộ dữ liệu thử nghiệm

SỐ THỨ TỰ	DỮ LIỆU VÀO	KẾT QUẢ RA
1	2 1 2	3
2	3 10 11 12	15

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

Bài 5

THỰC HÀNH KĨ THUẬT QUAY LUI GIẢI BÀI TOÁN XẾP HẬU

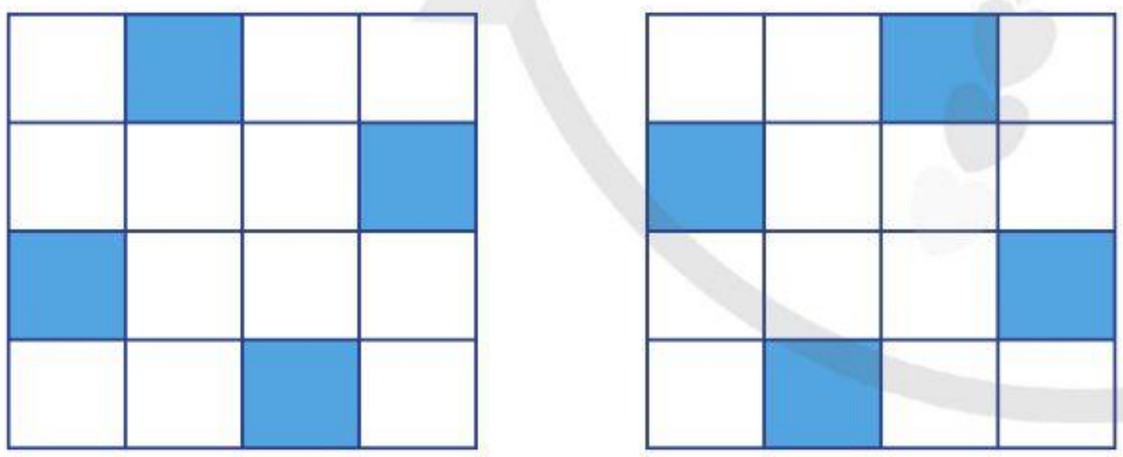
Học xong bài này, em sẽ:

Hiểu được các bước giải bài toán sử dụng kĩ thuật quay lui.

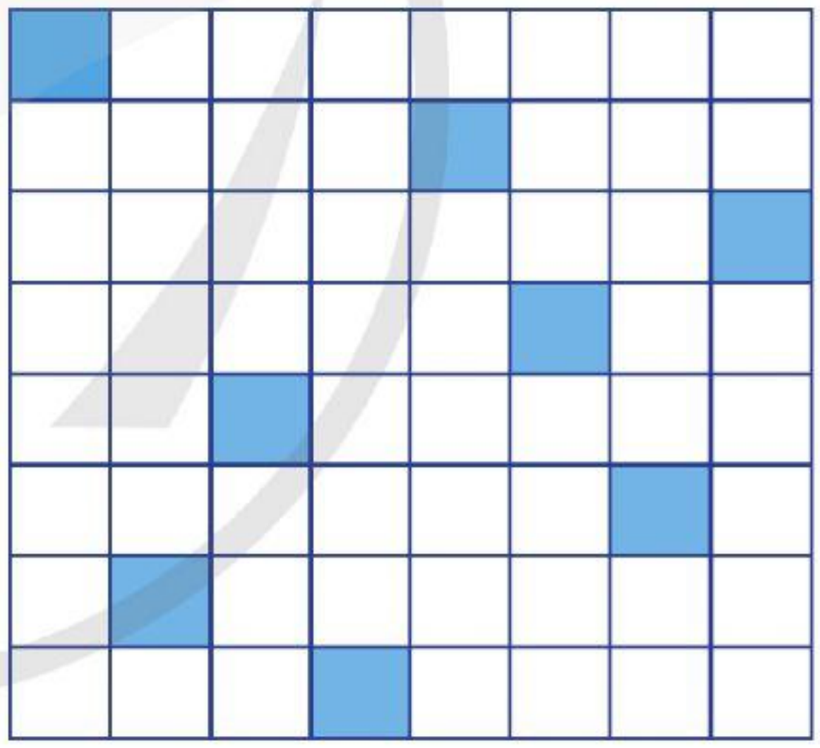
Bài toán *Xếp hậu*

Cần đặt n quân hậu lên bàn cờ vua kích thước $n \times n$, sao cho không có hai quân hậu nào tấn công nhau (tức là không có hai quân hậu nào cùng hàng, cùng cột hoặc cùng đường chéo).

Ví dụ 1: Trên bàn cờ 4×4 có tất cả hai cách đặt 4 quân hậu thoả mãn (*Hình 1*), các ô được tô màu là vị trí đặt quân hậu.



Hình 1. Hai cách đặt thoả mãn 4 quân hậu trên bàn cờ 4×4



Hình 2. Một cách đặt thoả mãn 8 quân hậu trên bàn cờ 8×8

Ví dụ 2: Có tất cả 92 cách đặt 8 quân hậu lên bàn cờ vua kích thước 8×8. *Hình 2* là một cách đặt thoả mãn.

Gợi ý:

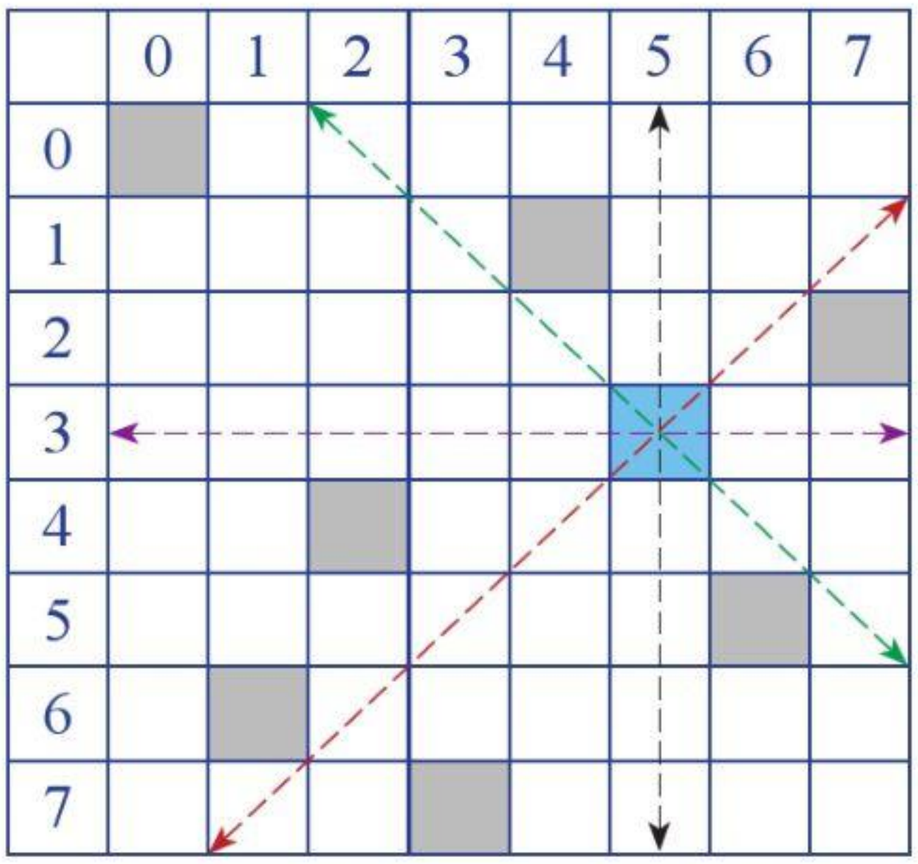
Bước 1. Thiết kế thuật toán theo kĩ thuật quay lui.

Trên bàn cờ, đánh số các dòng từ 0 đến $n - 1$ theo chiều từ trên xuống dưới, các cột đánh số từ 0 đến $n - 1$ theo chiều từ trái sang phải. Ô nằm giao của hàng i ($0 \le i \le n - 1$) và cột j ($0 \le j \le n - 1$) gọi là ô (i, j).

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Nhận thấy, mỗi hàng có đúng một quân hậu, mỗi cột có đúng một quân hậu nên lời giải bài toán có thể được biểu diễn bằng một dãy $X = (x_0, x_1, \ldots, x_{n-1})$ là hoán vị của 0, 1, …, $n - 1$, trong đó, x_i là chỉ số cột của quân hậu trên hàng thứ i ($0 \leq i \leq n - 1$), nghĩa là quân hậu thứ i sẽ được đặt ở ô (i, x_i).

Ví dụ, với $n = 8$, một cách đặt 8 quân hậu thoả mãn như *Hình 3* thì $X = (0, 4, 7, 5, 2, 6, 1, 3)$.



Hình 3. Khả năng tấn công của quân hậu

1

Với $n = 4$ có hai cách đặt 4 quân hậu thoả mãn, em hãy chỉ ra hai dãy số biểu diễn lời giải của hai cách đó theo dãy x, hai dãy đó có đặc điểm gì?

Quân hậu đặt ở ô (i, j) sẽ tấn công các quân hậu đặt ở ô cùng hàng, cùng cột hoặc cùng đường chéo. Cụ thể, quân hậu đặt ở ô (i, j) sẽ tấn công các quân hậu đặt ở:

– Các ô cùng hàng là các ô (i, j'), với $0 \leq j' \leq n - 1$.

– Các ô cùng cột là các ô (i', j), với $0 \leq i' \leq n - 1$.

– Các ô cùng đường chéo từ trên bên trái hướng xuống dưới sang bên phải (như đường màu xanh lá cây trong *Hình 3*) là các ô $(i + 1, j + 1)$, $(i + 2, j + 2)$,…, hay các ô $(i - 1, j - 1)$, $(i - 2, j - 2)$,…

– Các ô cùng đường chéo từ dưới bên trái hướng lên bên phải trên (như đường màu đỏ trong *Hình 3*) là các ô $(i - 1, j + 1)$, $(i - 2, j + 2)$,…, hay các ô $(i + 1, j - 1)$, $(i + 2, j - 2)$,…

2

Em hãy đưa ra điều kiện để kiểm tra hai quân hậu đặt ở hai ô (u_1, v_1) và (u_2, v_2) tấn công nhau.

Dãy $X = (x_0, x_1, \ldots, x_{n-1})$ là một hoán vị của dãy $(0, 1, \ldots, n - 1)$ biểu diễn một cách đặt thoả mãn nếu với mọi $i \neq j$ thì hai điều kiện sau thoả mãn: $i - x_i \neq j - x_j$ và $i + x_i \neq j + x_j$.

Với cách mô tả lời giải bài toán bằng một dãy $X = (x_0, x_1, \ldots, x_{n-1})$ là hoán vị của dãy $(0, 1, \ldots, n - 1)$, trong đó x_i là chỉ số cột của quân hậu trên hàng thứ i ($0 \leq i \leq n - 1$), ta có thể giải quyết bài toán bằng cách sử dụng kĩ thuật quay lui để liệt kê tất các hoán vị của dãy $(0, 1, \ldots, n - 1)$. Với mỗi hoán vị đó, tiến hành kiểm tra để chọn nghiệm.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bước 2. Xây dựng chương trình giải bài toán bằng kĩ thuật quay lui.

3

Tìm hiểu chương trình giải quyết bài toán xếp hậu bằng kĩ thuật quay lui trong *Hình 4*, giải thích ý nghĩa các hàm **`check`**, **`printSolution`**, **`permutation`** và chạy thử nghiệm với $n = 4, 5, 8$ và 10.

File Edit Format Run Options Window Help

```
1  def check(x):
2      for i in range(n):
3          for j in range(i+1,n):
4              if (i + x[i] == j + x[j]) or (i - x[i] == j - x[j]):
5                  return False
6      return True
7
8  def printSolution(x):
9      global cnt
10     cnt = cnt + 1
11     print("Cách thứ", cnt)
12     print(x)
13
14 def permutation(i):
15     for v in range(n):
16       if not (v in x):
17         x.append(v)
18         if i == n - 1:
19           if check(x):
20             printSolution(x)
21         else:
22           permutation(i + 1)
23         x.pop()
24
25 n = int(input())
26 x = []
27 cnt = 0
28 permutation(0)
```

Hình 4. Chương trình giải bài toán xếp hậu

Bài toán *Xếp hậu mở rộng*

Trên bàn cờ vua kích thước $n \times n$ có một ô (u, v) bị cấm, cần đặt n quân hậu lên bàn cờ sao cho không có hai quân nào tấn công nhau và không có quân nào đặt vào ô (u, v) bị cấm.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Bài 6

DỰ ÁN: XÂY DỰNG CHƯƠNG TRÌNH SỬ DỤNG KĨ THUẬT DUYỆT

1 Mục đích của dự án học tập

Dự án nhằm mục đích giúp học sinh khám phá, tìm hiểu xây dựng chương trình giải quyết các bài toán gần gũi với học sinh sử dụng kĩ thuật duyệt. Sau khi hoàn thành xong dự án, học sinh có khả năng:

– Vận dụng kiến thức đã học để giải quyết các vấn đề cụ thể.

– Tìm kiếm, khai thác, học hỏi mở rộng kiến thức từ các nguồn tài nguyên về lập trình trên Internet.

– Làm việc nhóm.

2 Yêu cầu chung

– Lớp chia thành 4 nhóm, các nhóm độc lập. Mỗi học sinh tham gia một nhóm, việc tham gia nhóm dự án nào là do học sinh lựa chọn, sau đó thầy, cô giáo điều chỉnh để đảm bảo cân đối số học sinh tham gia trong mỗi dự án.

– Mỗi nhóm lập trưởng nhóm và thảo luận tự chọn bài toán để xây dựng chương trình.

– Thời gian thực hiện dự án là một tuần. Các nhóm sẽ thực hiện tạo sản phẩm ngoài giờ lên lớp. Thầy, cô giáo hướng dẫn và lập kế hoạch một tiết trên lớp, sau khi các nhóm thực hiện xong dự án, 2 tiết trên lớp được dùng cho các nhóm báo cáo kết quả, mỗi nhóm 15 phút.

3 Một số hướng dẫn và gợi ý thực hiện dự án

a) Lập kế hoạch

Bước 1. Xác định mục tiêu dự án.

– Nhóm thảo luận lựa chọn, thống nhất bài toán.

– Các nhóm có thể tham khảo lựa chọn một trong các bài toán sau:

Bài toán 1. Biểu diễn số

Biết rằng một số tự nhiên n ($n \leq 1\,000$) luôn tồn tại số nguyên dương k ($k \leq 20$) và một cách điền dấu cộng hoặc dấu trừ vào các vị trí dấu * dưới đây để nhận được một đẳng thức đúng:

$$*1^2 * 2^2 * \cdots * k^2 = n$$

Đọc bản mới nhất trên hoc10.vn Bản mẫu

Với $n = 0$, ta có: $+ 1^2 + 2^2 - 3^2 + 4^2 - 5^2 - 6^2 + 7^2 = 0$.

Hãy viết chương trình nhập vào một số tự nhiên n ($n \leq 1\ 000$), đưa ra một cách biểu diễn thoả mãn.

Bài toán 2. Knock out chào 2022

Sử dụng lần lượt 9 chữ số từ 1 đến 9 cùng các phép toán cộng, trừ, nhân, chia để tạo ra các đẳng thức có kết quả là 2 022. Ví dụ, 1 234 + 5 − 6 + 789 = 2 022 là một cách thoả mãn.

Trong Python cung cấp hàm `eval()`, đầu vào là xâu biểu diễn một biểu thức, hàm sẽ trả về giá trị của biểu thức đó. Ví dụ, `eval('1234 + 5 - 6 + 789')` sẽ trả về 2 022.

Bài toán 3. Số siêu nguyên tố

Số siêu nguyên tố là số nguyên tố mà khi bỏ một số tuỳ ý các chữ số bên phải của nó thì phần còn lại vẫn tạo thành một số nguyên tố. Ví dụ, 2 333 là số siêu nguyên tố có 4 chữ số vì 233, 23, 2 cũng là các số nguyên tố.

Cho số nguyên dương N ($0 < N < 10$), đưa ra các số siêu nguyên tố có N chữ số cùng số lượng của chúng. Ví dụ, với $N = 4$, có 16 số: 2 333, 2 339, 2 393, 2 399, 2 939, 3 119, 3 137, 3 733, 3 739, 3 793, 3 797, 5 939, 7 193, 7 331, 7 333, 7 393.

Bài toán 4. Mã đi tuần

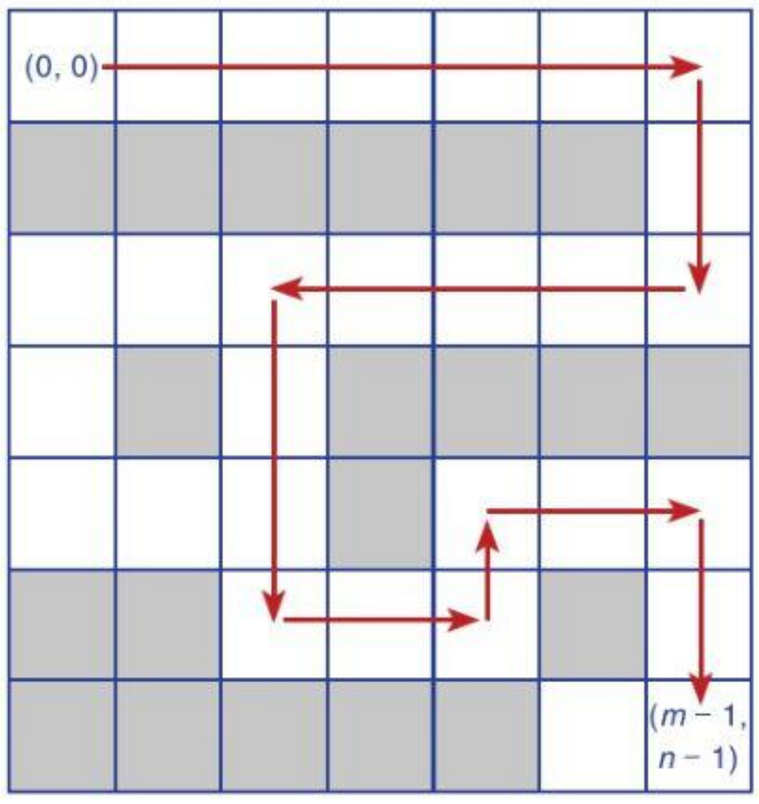
Cho bàn cờ $n \times n$ ô, tìm cách di chuyển một quân mã xuất phát từ một ô bất kì, mã di chuyển theo luật cờ vua và đi qua tất cả các ô, mỗi ô qua đúng một lần. Ví dụ, *Hình 1* là một phương án với $n = 5$.

1	24	13	18	7
14	19	8	23	12
9	2	25	6	17
20	15	4	11	22
3	10	21	16	5

Hình 1. Một cách di chuyển thoả mãn của quân mã trên bàn cờ 5 × 5

Bài toán 5. Tìm đường trong mê cung

Một mê cung được biểu diễn bằng một bảng số kích thước $m \times n$. Các hàng của bảng được đánh số từ 0 đến $m - 1$, các cột của bảng được đánh số từ 0 đến $n - 1$. Ô nằm giao giữa hàng i và cột j là ô (i, j) mô tả phòng (i, j). Trong mê cung có một số phòng cấm không được phép di chuyển vào. Xuất phát tại phòng (0, 0), người chơi cần tìm một đường đi di chuyển tới phòng $(m - 1, n - 1)$ (*Hình 2*).



Hình 2. Ví dụ một mê cung và một đường đi thoả mãn

Bước 2. Lập danh sách các công việc chính, thời gian thực hiện, hoàn thành.

Bước 3. Dự kiến sản phẩm.

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

b) Thực hiện dự án

Bước 1. Thiết kế thuật toán.

Bước 2. Sử dụng các lệnh trong Python để viết chương trình.

Bước 3. Thử nghiệm, kiểm thử để sửa lỗi, hiệu chỉnh chương trình.

c) Báo cáo kết quả

– Trong thời gian hai tiết, tiến trình thực hiện như trong *Bảng 1*.

Bảng 1. Nội dung, nhiệm vụ cần thực hiện

SỐ THỨ TỰ	NỘI DUNG, NHIỆM VỤ	NGƯỜI THỰC HIỆN	THỜI GIAN VÀ ĐỊA ĐIỂM	CẦN CHUẨN BỊ
1	Các nhóm báo cáo kết quả và tự đánh giá	Tất cả các nhóm	Phòng học bộ môn	Máy tính và máy chiếu
2	Tranh biện giữa các nhóm, đánh giá chéo	Tất cả các nhóm	Phòng học bộ môn	Máy tính và máy chiếu
3	Giáo viên tổng kết, đánh giá kết luận	Thầy, cô giáo và học sinh	Phòng học bộ môn	Máy tính và máy chiếu

– Khi báo cáo, các nhóm cần trình bày chi tiết các nội dung sau:

+ Giới thiệu, trình chiếu sản phẩm.

+ Trình bày kết quả thực hiện dự án, các chức năng và cách thức triển khai.

+ Nhận xét, tự đánh giá về việc thực hiện dự án.

4 Tiêu chí đánh giá

Việc đánh giá sản phẩm và báo cáo dự án như trong *Bảng 2*.

Bảng 2. Tiêu chí đánh giá

TIÊU CHÍ	MÔ TẢ	ĐIỂM
Chương trình	– Chạy đúng đắn, ổn định – Hiệu quả – Tuỳ biến, mở rộng	4
Giao diện	Đẹp, hấp dẫn	2
Báo cáo	– Trình bày rõ ràng – Trả lời câu hỏi tốt	4

Điểm của mỗi thành viên dựa trên sự đóng góp trong dự án.

Đọc bản mới nhất trên hoc10.vn Bản mẫu

BẢNG GIẢI THÍCH THUẬT NGỮ

Thuật ngữ	Giải thích	Trang
bài toán con	bài toán cùng dạng với bài toán cha nhưng có kích thước đầu vào nhỏ hơn	11
chia	chia nhỏ bài toán ra thành các bài toán con cùng dạng và độc lập nhau	40
đệ quy có nhớ	phương pháp đệ quy dùng thêm biến nhớ ghi nhận lời giải của các bài toán con, nhờ đó có thể giảm thời gian tính toán do mỗi khi cần đến có thể tra cứu mà không phải giải lại những bài toán con đã được giải trước đó	15
hoán vị của *n* phần tử	một dãy có thứ tự của *n* phần tử, mỗi phần tử trong dãy xuất hiện đúng một lần	61
lặp vô hạn	lặp đi lặp lại không bao giờ dừng	17
phạm vi tìm kiếm	tập những ứng viên cho lời giải của bài toán	27
quay lui	loại bỏ thành phần lời giải ở bước hiện tại và lùi về bước trước đó để xét khả năng tiếp theo	55
kết hợp	kết hợp lời giải của các bài toán con thành kết quả của bài toán cha	40
trị	giải các bài toán con theo cách giống như bài toán cha	40
trường hợp cơ sở	trường hợp dừng của đệ quy	11

Đọc bản mới nhất trên hoc10.vn
Bản mẫu

NHÀ XUẤT BẢN ĐẠI HỌC SƯ PHẠM

Địa chỉ: Tầng 6, Toà nhà số 128 đường Xuân Thuỷ, quận Cầu Giấy, TP. Hà Nội

Điện thoại: 024.37547735

Email: nxb@hnue.edu.vn | **Website:** www.nxbdhsp.edu.vn

Chịu trách nhiệm xuất bản:

Giám đốc – Tổng biên tập: NGUYỄN BÁ CƯỜNG

Chịu trách nhiệm tổ chức bản thảo và bản quyền nội dung:

CÔNG TY CỔ PHẦN ĐẦU TƯ XUẤT BẢN – THIẾT BỊ GIÁO DỤC VIỆT NAM

Chủ tịch Hội đồng Quản trị: NGƯT NGÔ TRẦN ÁI

Tổng Giám đốc: VŨ BÁ KHÁNH

Biên tập:

TRẦN THỊ HIÊN

ĐẶNG XUÂN THO – ĐÀO ANH TIẾN

Thiết kế sách và minh hoạ:

PHẠM VŨ TOẢN – LÊ ANH TUẤN

Trình bày bìa:

TRẦN TIỂU LÂM – NGUYỄN MẠNH HÙNG

Sửa bản in:

TRẦN THỊ HIÊN – TRẦN THỊ THANH VÂN

Trong sách có sử dụng một số hình ảnh trên Internet. Trân trọng cảm ơn các tác giả.

Chuyên đề học tập Tin học 11 – Khoa học máy tính

Mã số: .....

ISBN: 978-604-54-8679-5

In ....., khổ 19 x 26,5cm, tại .....

Địa chỉ: .....

Số xác nhận đăng kí xuất bản: .....

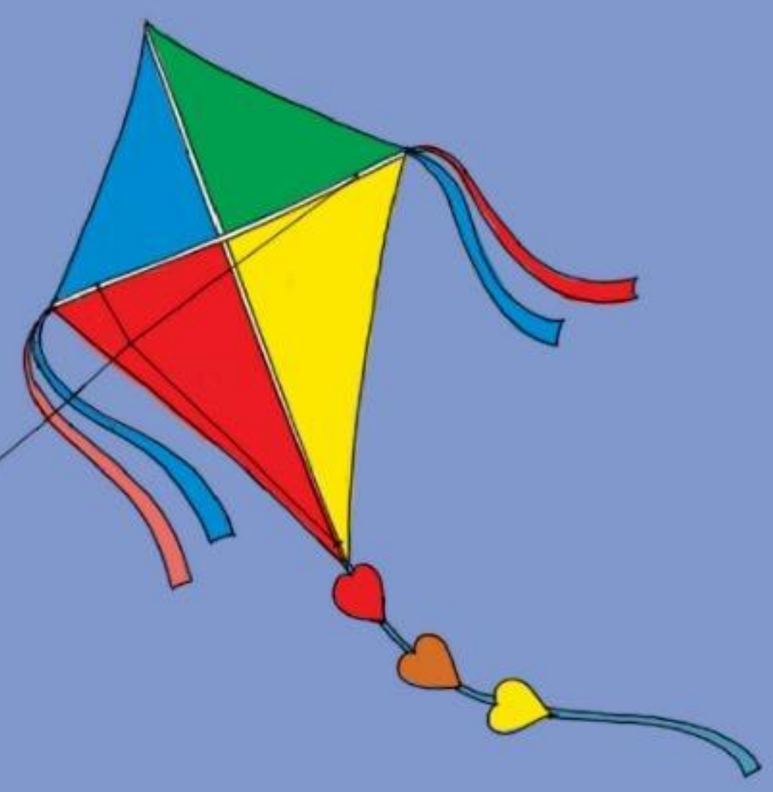
Quyết định xuất bản số: .....

In xong và nộp lưu chiểu tháng ..... năm 2023

Đọc bản mới nhất trên hoc10.vn

Bản mẫu

Mang cuộc sống vào bài học
Đưa bài học vào cuộc sống



Sách ***Chuyên đề học tập Tin học 11 – Khoa học máy tính*** sẽ hướng dẫn học sinh thực hành một số chiến lược thiết kế thuật toán cơ bản thông qua ba chuyên đề: *Thực hành thiết kế thuật toán theo kĩ thuật đệ quy*, *Thực hành thiết kế thuật toán theo kĩ thuật chia để trị* và *Thực hành thiết kế thuật toán theo kĩ thuật duyệt*. Các nội dung trong quyển sách này được thiết kế kĩ lưỡng, chú trọng việc thực hành, hoàn thiện sản phẩm, thông qua đó giúp học sinh phát triển tư duy máy tính, năng lực giải quyết vấn đề.

Sách do các nhà giáo giàu kinh nghiệm, nhiều tâm huyết trong lĩnh vực giáo dục tin học biên soạn.

SỬ DỤNG
TEM CHỐNG GIẢ

1. Quét mã QR hoặc dùng trình duyệt web để truy cập website bộ sách Cánh Diều: www.hoc10.com
2. Vào mục Hướng dẫn (www.hoc10.com/huong-dan) để kiểm tra sách giả và xem hướng dẫn kích hoạt sử dụng học liệu điện tử.

SÁCH KHÔNG BÁN

Đọc bản mới nhất trên hoc10.vn

Bản sách mẫu