**Bài 3:**

**Câu 1:**

Một chương trình con P (hàm hoặc thủ tục) được gọi là đệ quy nếu:

A. Trong quá trình thực hiện P có phần phải gọi đến chính P.

B. Trong quá trình thực hiện P phải gọi đến P hai lần.

C. Trong quá trình thực hiện P phải gọi đến P hai lần trở lên.

D. Trong quá trình thực hiện P không cần phải gọi đến P.

**Câu 2:**

Cấu trúc chính của một chương trình con đệ quy gồm:

A. Phần cơ sở và phần đệ quy.

B. Phần không cơ sở và phần không đệ quy.

C. Phần cơ sở và phần không có đệ quy.

D. Phần không cơ sở và phần đệ quy.

**Câu 3:**

Cho hàm đệ quy int power(int n)

{ if (n == 0) return 1;

else return 2 \* power(n-1); }.

Giá trị của power(3) được tính bằng:

A. 4

B. 10

C. 8

D. 6

**Câu 4:**

Hàm đệ quy dưới đây thực hiện chức năng gì?

int US(int a, int b) {

if (b == 0)

return a;

else

return US(b, a % b);

}

A. Tính tổng của a và b.

B. Tìm ước chung lớn nhất của a và b.

C. Tìm bội chung nhỏ nhất của a và b.

D. Tính hiệu của a và b.

**Câu 5:**

Hàm đệ quy dưới đây tính gì?

int sum(int n) {

if (n == 0)

return 0;

else

return n + sum(n - 1);

}

A. Tổng các số từ 1 đến n.

B. Tích các số từ 1 đến n.

C. Lũy thừa của n.

D. Tổng các số từ 1 đến n – 1.

**Câu 6:**

Hàm đệ quy sau thực hiện chức năng gì?

int countt(int n) {

if (n <= 0)

return 0;

else {

printf("%d ", n);

return countt(n - 1);

}

}

A. Đếm số lượng các số từ n đến 1 và in ra.

B. Tính tổng các số từ 1 đến n.

C. Tính giai thừa của số nguyên n.

D. In tất cả các số từ n, n-1, ... đến 1.

**Câu 7:**

Độ phức tạp của thuật toán đệ quy tính số hạng thứ n của dãy Fibonacci:

int fibo(int n) {

if (n <= 2)

return 1;

return fibo(n-1) + fibo(n-2);

}

Có kết quả bằng:

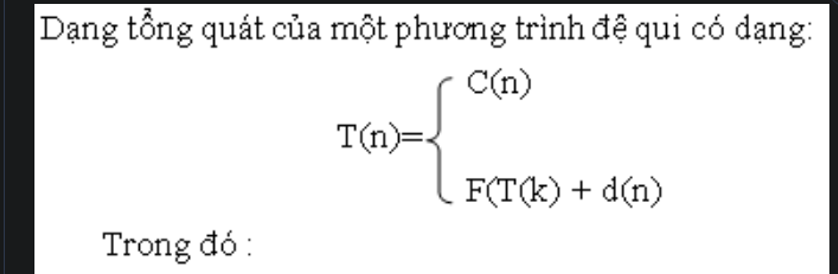
A. O(1)

B. O(2^n)

C. O(n!)

D. O(n^2).

***Câu 8:***



A.

C(n): là thời gian thực hiện chương trình.

F(T(k)) là một đa thức của các T(k).

d(n): là thời gian để phân chia bài toán.

B.

C(n): là thời gian thực hiện chương trình ứng với trường hợp đệ quy dừng.

F(T(k)) là một đa thức của các T(k).

d(n): là thời gian để phân chia bài toán và tổng hợp các kết quả.

**Câu 9:**

Cho hàm đệ quy:

int US(int a, int b) {

if (b == 0)

return a;

else

return US(b, a % b);

}

Giá trị của US(28, 20) được tính bằng:

A. 2

B. 4

C. 6

D. 8.

**Câu 10:**

Cho hàm đệ quy:

int sum(int n) {

if (n == 0)

return 0;

else

return n \* n + sum(n - 1);

}

Gọi hàm sum(3) trả về giá trị:

A. 20

B. 30

C. 14

D. 16.

**Câu 11:**

Độ phức tạp của thuật toán đệ quy thực hiện tìm kiếm giá trị x trong dãy có n phần tử theo phương pháp nhị phân:

int bsearch(int a[], int L, int R, int x) {

if (L > R)

return -1;

int k = (L + R) / 2;

if (x == a[k])

return k;

if (x > a[k])

return bsearch(a, k + 1, R, x);

return bsearch(a, L, k - 1, x);

}

Có kết quả bằng:

A. O(n log n)

B. O(log n)

C. O(n)

D. O(n^2).

**Câu 12:**

Thuật toán có phương trình đệ quy sau, (T(1)=1):

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

Được giải theo định lý Master có nghiệm là:

A. O(n)

B. O(n^2)

C. O(log n)

D. O(n log n).

**Câu 13:**

Thuật toán có phương trình đệ quy sau, (T(1)=1):

T(n) = 16T(n/4) + n

Được giải theo định lý Master có nghiệm là:

A. O(n)

B. O(n^2)

C. O(n log n)

D. O(log n).

**Câu 14:**

Thuật toán có phương trình đệ quy sau, (T(1)=1):

T(n) = 8T(n/2) + n

Được giải theo định lý Master có nghiệm là:

A. O(n^3)

B. O(n^2)

C. O(n^2 log n)

D. O(n).

**Câu 15:**

Với phương trình đệ quy T(n) = 8T(n/2) + n^3, hãy xác định bộ giá trị của (a, b, k) và nghiệm tính được theo định lý Master:

A. (8, 2, 3); O(n log n^2)

B. (8, 1, 3); O(n^3)

C. (8, 2, 3); O(n^3 log n)

D. (8, 1, 3); O(log n^3).

**Câu 16:** Khi đánh giá độ phức tạp của chương trình con đệ quy tính n! (n>=0), phương trình đệ qui được thành lập

có dạng:

A: c1 là hằng số

C1 nếu n=0

T(n-1)+T(n-2) nếu n>0

B: c1, c2 là hằng số

C1 nếu n=0

2T(n-1)+C2 nếu n>0

C 2T(n/2) + C2 nếu n>0

D T(n-1)+C2 nếu n>0

C1, C2 là các hằng số

**Câu 17:** Giá trị của power (2, 3) với đoạn chương trình sau, được tính bằng

int power(int base, int exp) {

if (exp = = 0) return 1;

else return base \* power(base, exp-1);

}

A 6

B 8

C 4

D 2

**Câu 18:** Cho Hàm đệ quy

int fibo(int n) {

if (n <= 2) return 1;

else return fibo(n-1) + fibo(n-2);

}

giá trị của fibo(5) được tính bằng:

A 8

B 3

C 13

D 5

**Câu 19:** Thuật toán có phương trình đệ quy sau, (T(1)=1)

T(n) = T(5n/9) + 1

được giải theo định lý Master có nghiệm là:

A O(n2)

B O(nlogn)

C O(logn)

D O(n)

**Câu 20:** Thuật toán có phương trình đệ quy sau, (T(1)=1)

T(n)=3T(n/3)+n

được giải theo định lý Master có nghiệm là:

A O(n2)

B O(logn)

C O(nlogn)

D O(n)

**Câu 21:** Thuật toán có phương trình đệ quy sau, (T(1)=1)

T(n)=4T(n/2)+n2

được giải theo định lý Master có nghiệm là:

A O(n2)

B O(nlogn)

C O(n2logn)

D O(n)

**Câu 22:**

Độ phức tạp của thuật toán đệ quy thực hiện sắp xếp các phần tử của một dãy có

𝑛 phần tử theo chiều tăng dần bằng phương pháp Quicksort trong trường hợp tốt nhất được xác định bởi phương trình sau:

Quicksort (int a[], int L, int R) {

k = Partition(a, L, R);

Quicksort(a, L, k-1);

Quicksort(a, k, R);

}

Độ phức tạp của thuật toán trong trường hợp tốt nhất là:

1. 𝑂(𝑛)
2. O(nlogn)
3. O(n^2)
4. O(2^n)

Câu 23: Phần cơ sở của thuật toán đệ qui bao gồm

A

các trường hợp dừng mà có thể trực tiếp

giải quyết được ngay

B

các trường hợp chưa dừng lại mà có thể

giải quyết trong trường hợp đặc biệt

C

các trường hợp chưa dừng nhưng có thể

trực tiếp giải quyết được

D

các trường hợp dừng nhưng chưa thể

trực tiếp giải quyết được

**Câu 23: Đặc điểm của chương trình con đệ qui :**

A

1. Trong chương trình con đệ quy có lời gọi đến chính nó

2.Mỗi lan có loi goi thì kích thuớc cua bai toan đa thu nho hơn trướ

3.Có một trường hợp đặc biệt và gọi đệ qui được tiếp tục

B

1. Trong chương trình con đệ quy có lời gọi đến chính nó

2.Mỗi lan co loi goi thì kích thuoc cua bai toan được thu nho đi một

nứa

3.Có một trường hợp đặc biệt, là trường hợp dừng của thuật toán và gọi

đệ qui cũng kết thúc

C

1. Trong chương trình con đệ quy có lời gọi đến chính nó

2.Mỗi lần có lời gọi thì kch thước cua bai toán được thu nhỏ đi một

nửa

3.Có một trường hợp đặc biệt và gọi đệ qui được tiếp tục

D

1. Trong chương trình con đệ quy có lời gọi đến chính nó

2.Mỗi lần có lời gọi thì kích thước của bài toán đã thu nhỏ hơn trước

3.Có một trường hợp đặc biệt, là trường hợp dừng của thuật toán và gọi

đệ qui cũng kết thúc

**câu 24**: Hàm đệ quy facto được định nghĩa như sau:

int facto(int n) {

if (n == 0) return 1;

else return n \* facto(n-1);

}

A

Bình phương của

B

Tổng các số từ 1 đến n

C

Trung bình cộng của các số từ 1 đến n

D

Giai thua cua n

**Câu 25**: Hàm đệ quy sau thực hiện chức năng gì?

int sum(int n) {

if (n <= 0) return 0;

else

if (n % 2 = = 0) return n + sum(n- 2);

else return sum(n - 1);

A

Tính tổng các số nguyên tố từ 1 đến n

B

Tính tổng các số chẵn từ 1 đến n

C

Tính tổng tất cả các số từ 1 đến n

D

Tính tổng các số lẻ từ 1 đến n

**Bài 1**

Câu 1: Thuật toán là

A

một day các bước, mỗi bước mô tả chính xác các phép

toán hoặc hành động cần thực hiện để giải quyết một

vấn đề

B

một day hữu hạn cac bước, mỗi bước mô ta chính xác

các phép toán hoặc hành động cần thực hiện để giải

quyết một vấn đề

C

một day hữu hạn cac bước, mô ta chính xac các phep

toán hoặc hành động để giải quyết một vấn đề

D

một day hữu hạn cac bưoc, mo ta chính xac các phep

toán hoặc hành động cần thực hiện để giải quyết một

vấn đề

câu2 : Giải thuật là

A

cách giải quyết bài toán cho kết quả gần đúng (chấp

nhận được) đỡ phức tạp và có hiệu quả hơn

B

cách giải quyết bài toán cho kết quả có hiệu quả hơn

C

cách giải quyết bài toán đảm bảo các đặc trưng của

thuật toán

D

cách giải quyết bài toán cho kết quả đúng

**câu 3:** Tính hữu hạn của thuật toán là

A

thuật toán bao giờ cũng phải dừng lại sau một số hữu

hạn bước

B thuat toan sẽ dừng lại sau mot số lần lặp các bước

C

thuật toán sẽ dừng lại sau một số bước thực hiện

D

thuật toán bao giờ cũng phải dừng lại sau một số vô

hạn bước thực hiện

**câu 4:** Giá về thời gian trên máy Turing là:

A

Thời gian để thực hiện các bước chuyển hình trạng từ

hình trạng đầu đến hình trạng cuối

B

Thời gian để thực hiện bước chuyển hình trạng đầu

C

Thời gian để thực hiện các bước chuyển hình trạng

trung gian

D

Thời gian để thực hiện bước chuyển hình trạng cuối

**Câu5**: Với máy xử lý thuật toán bằng ngôn ngữ tựa ALGOL, đơn vị nhớ là:

A

Một cho nhớ để chứa một dữ liệu ra

B

Một chỗ nhớ để chứa một dữ liệu

C

Một chỗ nhớ để chứa một dữ liệu vào

D

Một chỗ nhớ để chứa một kí hiệu

**Câu 6 :** Với bài toán: Xác định giá trị lớn nhất trong dãy có n số nguyên X=(x1, X2, ... ,Xn}, n là số nguyên dương. Hãy xác định kích thước của bài toán theo quan niệm thứ nhất:

A n2

B n+1

C nlogn

D n

**Câu 7:** Xác định Input, Output cho bài toán tìn kiếm tuần tự giá trị k trong dãy n số nguyên khác nhau X1, X2, ... , Xn

A

Input : số nguyên dương n, dãy n số nguyên khác nhau

X1, X2, ... , Xn, Số nguyên k

Output : Vị trí i mà xi = k hoac thông báo không tìm thấy

số nguyên k trong dãy

B

Input : day n số nguyên khác nhau X1, X2, ... , Xn, Số

nguyên k

Output : Vị trí i mà x = k hoac thông báo không tìm thấy

số nguyên k trong dãy

C

Input : day n số nguyên khác nhau x1, X2, ... , Xn, Số

nguyên k

Output : Vị trí i mà xị = k

D

Input: số nguyên dương n, day n số nguyên khác nhau

X1, X2, ... , Xn

Output : Vị trí i mà Xi = k hoac thông báo không tìm thấy

số nguyên k trong dãy

câu 8 : Xác định Input của bài toán : Hoán đổi giá trị của 2 biến số nguyên x và y và dùng biến trung gian số nguyên z

A

Hai biến số nguyên y, z

B

Hai biến số nguyên x, z

C

Hai biến số nguyên x, y

D

Ba biến số nguyên x, y, z

**Câu 9**: Phương pháp giả mã dùng để biểu diễn thuật toán là

A

mượn một ngon ngu lap trình nao đo đe viết chương

trình

B

mượn các cú pháp của một ngôn ngữ lập trình nào đó

để thể hiện thuật toán

C

dùng cấu trúc của một ngôn ngữ lập trình bậc cao để

viết chương trình

D

dùng cấu trúc của ngôn ngữ lập trình bậc thấp để mô

tả thuật toán

**câu 10**: Khi biểu diễn thuật toán bằng lưu đồ khối (sơ đồ khối), hình chữ nhật có ý nghĩa gì?

A

Thực hiện thao tác kiểm tra du liệu theo điều kiện để

phân nhánh thuật toán

B

Thực hiện thao tác nhập và xuất dữ liệu

C

Thực hiện thao tác xử lý du liệu (gán, các phép tính cơ

bản)

D

Thực hiện thao tác ghi và nhập dữ liệu

**Câu 11**: Với bài toán : Xác định giá trị lớn nhất trong dãy n số nguyên X={x1, X2, ... ,Xn}, n là số nguyên dương.

Hãy chọn biểu diễn Input, Output đúng :

A

Input : Dãy số nguyên X= {X1, X2, ... , Xn}, n nguyên dương.

Output: Tìm giá trị lớn nhất Max của dãy X

B

Input : Dãy số nguyên X= {x1, X2, ... , Xn}

Output: Tìm số giá trị lớn nhất

C

Input : Day số nguyên X= {x1, X2, ... , Xn}, n

Output: Tìm giá trị lớn nhất

D

Input : Day số nguyên X= {x1,X2, ... ,xn}

Output: Tìm gia trị lớn nhất Max của X

Câu 11 : Cho day số nguyên có n phần tử :X1, X2, ... , Xn. Neu thuật toán tìm thấy và đưa ra chỉ số thứ i đầu tiên thỏa man xi là số chan thì với

điều kiện nào thuật toán sẽ dừng :

A

X¡ là số chẵn

B

i>=n

C

Xi là số lẻ

D

i<n

câu 12: Cho day số nguyên có n phần tử X1, X2, ... , Xn và số nguyên k. Nếu thuật toán tìm thấy và đưa ra chỉ số thứ i đầu tiên thỏa mãn Xi =k

thì với điều kiện nào thuật toán sẽ dừng :

A

i>=n

B

Xi =k

C

i<n

D

Xj <>k

**Câu 13:** Độ phức tạp dữ liệu vào của bài toán theo quan niệm thứ nhất là

A

số lượng dữ liệu được xử lý của bài toán

B

số lượng dữ liệu đã được sử dụng vào của bài toán

số lượng dữ liệu trung gian của bài toán

D

số lượng dữ liệu vào của bài toán

câu 14: Lưu đồ khoi dùng để biểu diễn thuật toán là

A

một hệ thống các nút (nút giới hạn, nút thao tác, mũi

tên) được nối với nhau bởi các cung (mũi tên)

B

một hệ thống các nút (nút giới hạn, nút thao tác, nút

điều kiện) có hình dạng khác nhau theo qui ước, thể

hiện các chức năng khác nhau và được nối với nhau

bởi các cung (mũi tên)

C

một hệ thống các nút (nút giới hạn, nút thao tác, nút

điều kiện, mũi tên) thể hiện các chức năng khác nhau

và không được nối với nhau

D

một hệ thống các nút (nút giới hạn, nút điều kiện, mũi

tên) có hình dạng khác nhau theo qui ước, thể hiện các

chức năng khác nhau

**câu 15:** Với máy xử lý thuật toán bằng ngôn ngữ tựa ALGOL, đơn vị nhớ là:

A

Một chỗ nhớ để chứa một kí hiệu

B

Một chỗ nhớ để chứa một dữ liệu

C

Một chỗ nhớ để chứa một dữ liệu vào

D

Mot chỗ nhớ đe chứa một dữ liệu ra

Câu 16: Với bài toán: Xác định giá trị lớn nhất trong dãy n số nguyên X={x1, X2, ... ,Xn), n là số nguyên dương.

Hãy xác định kích thước của bài toán theo quan niệm thứ 2:

A

Σ[1og2xi]+log2n+n

B

Σ [1og2xi] +log2n+n+1

C

Σ [1og2 xi] +log211

D

Σ[logg2 xi]

i=1

câu 16: Biểu diễn thuật toán theo ngôn ngữ tự nhiên là

A

sử dụng ngôn ngữ thường ngày để lập chương trình

B

sử dụng ngôn ngữ chữ viết thường ngày

C

sử dụng ngôn ngữ chữ viết để ve thuật toán

D

sử dụng ngôn ngữ chữ viết thường ngày để liệt kê các

bước của thuật toán

câu 17 : Tính hiệu quả của thuật toán được đánh giá dựa trên các tiêu chuẩn:

A

Thời gian cần thiết để chạy chương trình

B

Dung lượng bộ nhớ của máy tính và thời gian chạy

chương trình

C

Dung lượng bộ nhớ cần có

D

Dung lượng bộ nhớ cần có và thời gian cần thiết để

chạy chương trình

cau 18: Trong biểu diễn một bài toán trên máy tính, Input là

A

Các dữ liệu vào của bài toán

B

Dữ liệu trong tính toán của bài toán

C

Dữ liệu trung gian của bài toán

D

Mot Dữ liệu vào của bài toán

**Bài 2**

**Câu 1** : Khi nói thời gian thực hiện của một chương trình là T(n) = Cn (C là hằng số) thì có nghĩa là

A

chương trình đó cần Cn dữ liệu vào

B

chương trình đó cần Cn dữ liệu tính toán

C

chương trình đó cần Cn chỉ thị thực thi

D

chương trình đó cần Cn dữ liệu ra

**caau2:** Cách đánh giá thời gian thực hiện thuật toán độc lập với máy tính và các yếu tố liên quan tới máy tính sẽ dẫn tới khái niệm gọi là

A

độ phức tạp của dữ liệu vào thuật toán

B

độ phức tạp không gian của thuật toán

C

độ phức tạp dữ liệu ra của thuật toán

D

độ phức tạp tính toán của thuật toán

**câu 3:** Nếu T1(n) và T2(n) là thời gian thực hiện 2 chương trình P1, P2 và T1(n)=O(f(n)), T2(n)=O(g(n)). Thời gian thực hiện của 2 chương trình

nối tiếp nhau là:

A

T(n)=max(f(n),g(n))

B

T(n)=O(f(n)\*g(n))

C

T(n)=O(max(f(n),g(n))

D

T(n)=O(f(n))\*O(g(n))

**Câu 4** Nếu độ phức tạp của lenh là O(1) thì độ phức tạp của đoạn chương trình sau:

for (i=1 ; i <= n ; i++)

for (j=1 ; j <= n ; j++)

{lệnh

được xác định bằng

A

O(nlogn)

B

O(1)

C

O(n)

D

0(n^2)

**Câu 5:** Nếu độ phức tạp của lệnh 1 và lệnh 2 đều là O(1) thì độ phức tạp của đoạn chương trình sau:

for (i=1 ; i <= n ; i++) {lệnh 1}

for (j=1 ;j <= m ; j++) {lệnh 2}

được xác định bằng:

A

0(m)

B

0(n\*m)

C

O(max(n,m))

D

0(n)

**Câu 6 :** Nếu độ phức tạp của lệnh là O(1) thì độ phức tạp của đoạn chương trình sau :

for (i=1 ; i <= n ; i++) {

for (j=1 ; j <= m; j++) {lệnh}

for (k=1 ; k <= h ; k++) {lệnh}

được xác định bằng:

A

O(n+m\*h)

B

O(n\*max(m,h))

C

O(n\*(m+h))

D

O(n\*m\*h)

**Câu 7**: Nếu T1(n) và T2(n) là thời gian thực hiện 2 đoạn chương trình P1, P2 và T1(n)=O(f(n)), T2(n)=O(g(n)). Thời gian thực hiện 2 đoạn

chương trình lồng nhau:

A

T(n)=f(n)\*g(n)

B

T(n)=O(max(f(n),g(n))

C

T(n)=O(f(n)\*g(n))

D

T(n)=max(f(n),g(n))

**Câu 8:** Khi đánh giá độ phức tạp của thuật toán, hàm nào dưới đây có tốc độ tăng trưởng chậm nhất?

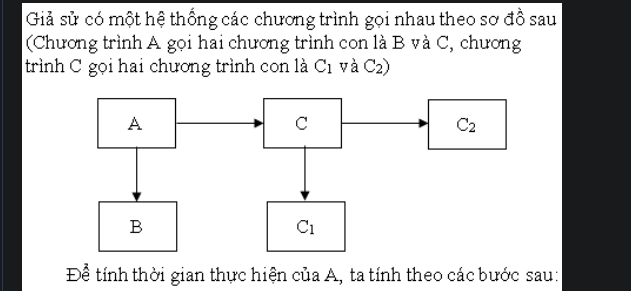
A O(nlogn)

B O(n)

C 0(n2)

D O(logn)

**Câu 9 :**



A

1. Tính thời gian thực hiện của B, C1

2. Tính thời gian thực hiện của C2

3. Tính thời gian thực hiện của A

B

1. Tính thời gian thực hiện của C1 và C2

2. Tính thời gian thực hiện của B

3. Tính thời gian thực hiện của A

C

1. Tính thời gian thực hiện của B

2. Tính thời gian thực hiện của C1, C2

3. Tính thời gian thực hiện của A

D

1. Tính thời gian thực hiện của B, C1 và C2

2. Tính thời gian thực hiện của C

3. Tính thời gian thực hiện của A

**Câu 10:** Xác định độ phức tạp cho đoạn chương trình sau:

S = 1; p = 1;

for (i=1; i <= n; i++) {

p=p\*x/ i;

S = s+ p;

}

A

O(logn)

B

0(n)

C

O(nlogn)

D

0(n2)

**Câu 11:** Xác định độ phức tạp cho đoạn chương trình sau:

if (m<n) p = m; else p = n;

for (i=0; i <= p; i++)

c[i]=a[i] + b[i];

if (p<m)

for (i=p+1; i <= m; i++) c[i] = a[il;

else

for (i=p+1; i <= n; i++) c[i] = b[i];

while (p>0 && c[p] = 0) p = p-1;

A

0(m\*n)

B

O(m+n)

C

O(max(m,n))

D

O(logmn)

**Câu 12:** Xác định độ phức tạp cho đoạn chương trình sau:

for (int i = 0; i < n; i++) {

for (int j = 0; j < n / 2; j++) {

for (int k = 0; k < 10; k++) {

printf("%d %d %d\n", i, j, k);

}

}

}

A

O(n2\*10)

B

0(n2)

C

O(n logn)

D

O(n)

**Câu 13** Xác định độ phức tạp cho đoạn chương trình sau:

for (int i = 0; i < n; i++) {

for (int j = 0;j < n; j++) {

for (int k = 0; k < logn; k++) {

printf("%d %d %d\n", i, j, k);}} }

A

O(n log n)

B

O(n2 log n)

C

O(logn)

D

0(n3)

**Câu 14:** Chi phí thời gian của một quá trình tính toán là

A

thoi gian cần thiết đe kiểm tra qua trình tính toán

B

thời gian cần thiết để thực hiện một qua trình tính toán

C

thời gian cần thiết để thiết kế thuật toán

D

thời gian cần thiết để xây dựng thuật toán

**câu 15** : Với máy xử lý thuật toán bằng ngôn ngữ tựa ALGOL, giá về thời gian là:

A

Số phép tính số học

B

Số phép tính quan hệ

C

Số phép tính logic

D

Số phép tính căn bản

**Câu 16:** Nếu độ phức tạp của lệnh là O(1) thì độ phức tạp của đoạn chương trình sau :

for (i=1 ; i <= n ; i++) {lệnh}

được xác định bằng:

A

O(1)

B

O(logn)

C

0(n)

D

O(nlogn)

**Câu 17**: Xác định độ phức tạp cho đoạn chương trình sau:

int s=0;

for (int i=1; i <= n; ++i)

for (int j=1; j≤i; ++j)

S=S+1;

printf("%d \n", s);

A

0(n2)

B

0(logn)

C

O(n)

D

O(nlogn)

**Câu 18:**

Xác định độ phức tạp cho đoạn chương trình sau:

int i = 0;

while (i < n) {

int j = i;

while (j < n) {

printf("%d %d\n", i, j);

j+=2;}

i++;}

A

0(n)

B

O(n log n)

C

O(n3)

D

O(n2)

**Câu 19:** Xác định độ phức tạp cho đoạn chương trình sau:

for (int i = 0; i < n; i++) {

for (int j = 0; j < i; j++) {

for (int k = j; k < n; k++) {

printf("%d %d %d\n", i, j, k);} } }

A

0(n)

B

0(n2)

C

O(n3)

D

O(n logn)

**Câu 20 :** Xác định độ phức tạp cho đoạn chương trình sau:

void BS(int all, int n) {

for(int i=0; i<n-1; i++) {

for(int j=n-1; j>i; j-) {

if (aljil<a[j-1]) {

int tg = aljl;a[i] = a[j-1];a[j-1] = tg;

for(int i=0; i<n; i++){ printf("%d \t",a[i]);}

A

O(n logn)

B

0(n2 logn)

C

0(n)

D

O(n2)

**Bài 4**

**Câu1:** Thuật toán được gọi là đa thức nếu

A

độ phức tạp về không gian trong trường

trung bình của nó là đa thức

B

độ phức tạp về không gian trong trường

xấu nhất của nó là đa thức

C

độ phức tạp về thời gian trong trường hợp

xấu nhất của nó là đa thức

D

độ phức tạp về thời gian trong trường hợp

tốt nhất của nó là đa thức

**câu2:** Thuật toán đơn định đa thức là

A

thuật toán đơn định có độ phức tạp trong

trường hợp trung bình là đa thức

B

thuật toán đơn định có độ phức tạp trong

trường hợp tốt nhất là đa thức

C

thuật toán đơn định có độ phức tạp là đa

thức

D

thuật toán đơn định có độ phức tạp là

trên đa thức (hàm mũ)

**câu** 3: Một bài toán thuộc lớp P nếu:

A

Nó có thể được giải quyết trong thời gian

đa thức bằng một thuật toán đơn định

B

Nó không được giải quyết bằng thuật

toán nào cả

C

Nó có thể được giải quyết trong thời gian

đa thức bằng một thuật toán không đơn

định

D

Nó không được giải quyết trong thời gian

đa thức

**câu 4**: Lớp NP bao gồm những bài toán:

A

Chưa tìm được thuật toán không đơn

định với độ phức tạp hàm mũ

B

Chưa tìm được thuật toán với độ phức

tạp đa thức nhưng chỉ ra được phương

pháp kiểm định nghiệm của nó (nếu có)

với thời gian hàm mũ

C

Chưa tìm được thuật toán đơn định với

độ phức tạp hàm mũ

D

Chưa tìm được thuật toán với độ phức

tạp đa thức nhưng chỉ ra được phương

pháp kiểm định nghiệm của nó (nếu có)

với thời gian đa thức

**câu 5:** Kiểm định nghiệm trong thời gian đa thức có nghĩa là:

A

Tìm kiếm nghiệm gần đúng trong thời

gian đa thức

B

Tìm kiếm nghiệm đúng trong thời gian đa

thức

C

Giải quyết bài toán trong thời gian đa

thức

D

Kiểm tra một nghiệm có đúng hay không

trong thời gian đa thức

**câu 6:** Mot thuat toan tìm số ước của một số nguyên dương N

1.Nhập N

2. dem=0;

3. for (i=1; i <= N; ++i)

4 if (N%i == 0)

5 dem=dem+1;

6. Xuất dem

Được đánh giá là:

A

Thuật toán đơn định, hàm mũ

B

Thuật toán đơn định, đa thức

C

Thuật toán không đơn định, hàm mũ

D

Thuật toán không đơn định, đa thức

**Câu 7**: Một thuật toán giải bài toán chiếc ba lô 0-1

For i := 1 to n do

Xi := CHOICE({0,1}); (phép toán lựa chọn một trong 2 giá trị)

if I xiai =B then SUCCESS

i-1

else FAILURE;

được đánh giá là:

A

Thuật toán không đơn định, hàm m

B

Thuật toán đơn định, hàm mũ

C

Thuật toán đơn định, đa thức

D

Thuật toán không đơn định, đa thức

**Câu 8**: Nếu một bài toán thuộc lớp NP nhưng không thuộc lớp P thì:

A

Lời giải của nó có thể tìm thấy trong thời

gian đa thức

B

Lời giải của nó không tồn tại

C

Lời giải của nó không thể được kiểm định

trong thời gian đa thức

D

Lời giai của nó có được kiểm định trong

thời gian đa thức nhưng không thể tìm

được trong thời gian đa thức

**câu 9**: Thuật toán nào sau đây có khả năng giải quyết bài toán NP trong thời gian đa thức?

A

Không có lựa chọn nào đúng

B

Thuật toán không đơn địn

C

Thuật toán đơn định

D

Thuật toán heuristic

**Caau10;** Bài toán “xác định số nguyên tố", có thể được giải quyết trong thời gian đa thức bởi thuật toán đơn định

không?

A

Không

B

Có

C

Chỉ khi sử dụng thuật toán không đơn

định

D

Chỉ khi số nguyên tố rất nhỏ

**Câu 11**: Thuật toán trên máy xử lý thuật toán bằng ngôn ngữ tựa ALGOL là đa thức thì:

A

Thuật toán tương ứng trên máy Turing là

không đơn định

B

Thuật toán tương ứng trên máy Turing là

đa thức

C

Thuật toán tương ứng trên máy Turing

chưa chắc là đa thức

D

Thuật toán tương ứng trên máy Turing là

đơn định

**câu 12:** Bài toán Tìm đường đi ngắn nhất trong đồ thị có trọng số thuộc lớp nào?

A

NP

B

NP-Hard

C

Không thuộc P hoặc NP

D

P

**Câu 13**: Bài toán "tìm chu trình Euler trong một đồ thị", có thuộc lớp P?

A

Không

B

Chỉ khi sử dụng thuật toán không đơn

định

C

Chỉ khi đồ thị có số cạnh nhỏ

D

Có

**Câu 14**: Bài toán “kiểm tra đồ thị có chứa chu trình Hamilton", có thuộc lớp P?

A

Có

B

Chỉ khi sử dụng thuật toán không đơn

định

C

Không

D

Chỉ khi đồ thị có số cạnh rất nhỏ

**Câu 15:** Bài toan "xác định một số nguyên dương N có phải là số nguyên tố hay không" Có thuộc lớp P?

A

Chỉ khi số rất nhỏ

B

Chỉ khi sử dụng thuật toán không đơn

định

C

Không

D

Có

Câu 16:

**Bài 5**

**Câu 1: Cho hai bài toán A và B, A được gọi là "dẫn về được" B một cách đa thức nếu**

A

có một thuật toán đơn định đa thức để

giai bài toán B thì cũng có một thuật toán

đơn định đa thức khác để giải bài toán A.

B

có một thuật toán không đơn định đa

thức để giải bài toán B thì cũng có một

thuật toán không đơn định đa thức khác

để giải bài toán A

C

có một thuật toán đơn định để giải bài

toán B thì cũng có một thuật toán đơn

định khác để giải bài toán A

D

có một thuật toán đơn định đa thức để

giai bài toán A thì cũng có một thuật toán

đơn định đa thức khác để giải bài toán B.

**Câu 2: Nếu bài toán A “dẫn về được" bài toán B sau thời gian đa thức, thì**

A

bài toán B là trường hợp riêng của bài

toán A

B

bài toan A "khó hơn" bài toan B

C

bài toan A "kho bằng" bài toan B

D

bài toan A "dễ hơn" bài toán B

**câu 3: Khái niệm phép quy dẫn trong lý thuyết độ phức tạp là:**

A

Phương pháp để tăng cường độ chính

xác của thuật toán

B

Phương pháp cải thiện tốc độ của thuật

toán

C

Quy trình để giảm kích thước dữ liệu

D

Quy trình giải một bài toán bằng cách sử

dụng bài toán khác

**câu 4: Bài toán A được gọi là NP-Hard (NP- khó) nếu:**

A

Tồn tại thuật toán để giải bài toán A thì

kéo theo sự tồn tại thuật toán để giải một

bài toán trong NP

B

Tồn tại thuật toán đa thức để giải bài

toán A thì kéo theo sự tồn tại thuật toán

đa thức để giải mọi bài toán trong NP

C

Tồn tại thuật toán đa thức để giải bài

toán A thì kéo theo sự tồn tại thuật toán

đa thức để giai một bài toán trong NP

D

Tồn tại thuật toán để giải bài toán A thì

kéo theo sự tồn tại thuật toán để giải mọi

bài toán trong N

**câu 5: Nếu một bài toán thuộc lớp NPC, thì điều gì đúng?**

A

Nó thuộc lớp NP-Hard nhưng không

thuộc NP

B

Nó không thuộc lớp NP

C

Nó thuộc lớp NP và mọi bài toán trong

NP có thể quy dẫn đến nó

D

Nó không thể giải quyết được

**Câu 6: Bài toán mà đầu ra chỉ có thể là “Yes" hoặc "No" (Đúng/sai, chấp nhận/từ chối) được gọi là:**

A

Bài toán đếm

B

Bài toán quyết định

C

Bài toán tối ưu

D

Bài toán liệt kê

**Câu 7: Bài toán 3-SAT được phát biểu :**

A

Cho một công thức CNF, hỏi rằng có tồn

tại một bộ gia trị của các biến sao cho

biểu thức nhận giá trị TRUE hay không?

B

Cho một công thức 3-CNF, hỏi rang có

tồn tại một bộ giá trị của các biến sao

cho biểu thức nhận giá trị TRUE hay

không?

C

Cho một công thức CNF, hỏi rằng có tồn

tại một bộ giá trị của các biến sao cho

biểu thức nhận giá trị FALSE hay không?

D

Cho một công thức 3-CNF, hỏi rang có

tồn tại một bộ giá trị của các biến sao

cho biểu thức nhận giá trị FALSE hay

không?

**Câu 8: Khi bài toán A "dẫn về được" bài toán B sau thời gian đa thức, được hiểu là:**

A

Bài toan B "kho bằng" bài toán A

B

Bài toan B "khó hơn" bài toán A

C

Bài toán A "khó bằng" bài toán B

D

Bài toan A "khó hơn" bài toán B

**Câu 9: Để chứng minh bài toán B là NPC cần thực hiện:**

1.Chứng minh B thuộc NP;

2. Tìm bài toán A thuộc NP;

3. Chứng minh bài toán A quy dẫn về bài toán B

B

1.Chứng minh B thuộc NP;

2. Tìm bài toán A thuộc NP-Hard;

3. Chứng minh bai toan B quy dẫn về bai toan A

C

1.Chứng minh B thuộc NP;

2. Tìm bài toán A thuộc NPC;

3. Chứng minh bài toán A quy dẫn về bài toán B

D

1.Chứng minh B thuộc NP;

2. Tìm bài toán A thuộc N;

3. Chứng minh bài toan A quy dẫn về bài toán B

**Câu 10 Bài toán TSP (người du lịch) thuộc lớp nào:**

A

NP

B

P

C

NP-Hard

D

NPC

**Câu 11 Bài toán tìm chu trình Hamilton thuộc lớp nào:**

A NPC

B P

C NP-Hard

D NP

**Câu 12Bài toán Max-Cut thuộc lớp bài toán nào:**

A NPC

B P

C NP

D NP-Hard

C**âu 13 Bài toán Ba lô gia trị nguyên thuộc lớp bài toán nào:**

A : NP

B: P

C : NP-Hard

D

NPC

**Câu 14 : Bài toán tô màu đồ thị thuộc lớp bài toán nào:**

A NPC

B P

C NP-Hard

D NP

**Câu 15: Bài toan về be lớn nhat của đồ thị (MaxClique) thuộc lớp bài** **toán nào:**

A NP

B bNP-Hard

C NPC

D P

**Câu 16: Cho hai bài toan A và B, A được gọi là "dẫn về được" B một cách đa thức nếu**

A

có một thuật toán đơn định đa thức để

giai bài toán B thì cũng có một thuật toán

đơn định đa thức khác để giải bài toán A.

B

có một thuật toán không đơn định đa

thức để giải bài toán B thì cũng có một

thuật toán không đơn định đa thức khác

để giải bài toán A.

C

có một thuật toán đơn định đa thức để

giai bài toán A thì cũng có một thuật toán

đơn định đa thức khác để giải bài toán B.

D

có một thuật toán đơn định để giải bài

toán B thì cũng có một thuật toán đơn

định khác để giải bài toán A

**câu 17 : Một bài toán A có thể được quy dẫn về bài toán B nếu:**

A

Bai toan B là dễ hơn bài toán A

B

Bài toán A có thể được giải quyết bằng

cách giải bài toán B

C

Bài toán B có thể được giải quyết nhanh

hơn bài toán A

D

Bài toán A không thể giải quyết được

**Câu 18: Lớp NPC bao gồm những bài toán nào:**

A

Những bài toán không thể giải quyết

được

B

Những bài toán có thể giai quyết trong

thời gian đa thức

C

Những bài toán có thể giải quyết trong

thời gian đa thức bằng thuật toán không

đơn định

D

Những bài toan có thể kiểm định nghiệm

trong thời gian đa thức và mọi bài toán

trong NP có thể quy dẫn đến chúng

**Câu 19: Nếu bài toán A là NP-Hard, điều gì đúng?**

A

A thuộc lớp P

B

A không thể thuộc NP

C

A có thể giải quyết trong thời gian đa

thức

D

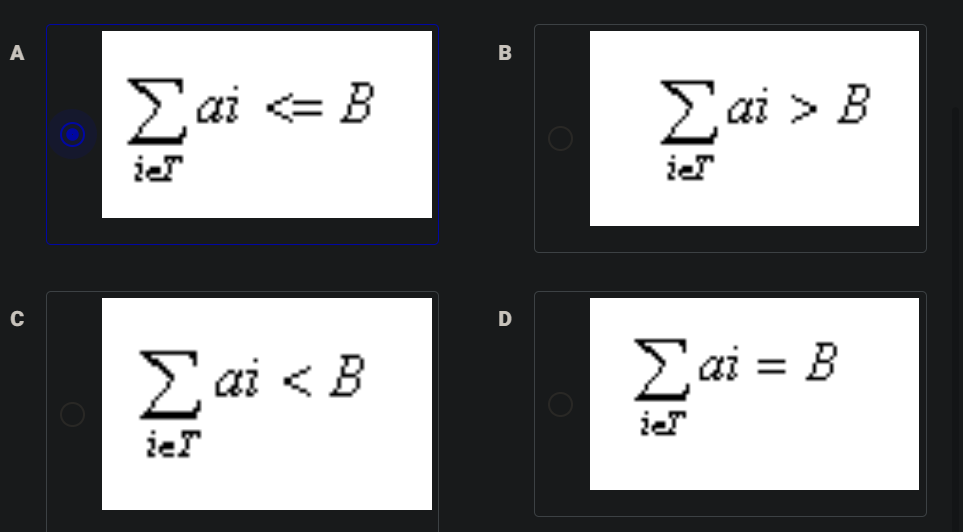
Bài toán trong NP có thể quy dẫn đến bài

toán A

**Câu 20:** Cho 1 ba lô có trọng lượng là B, n đồ vật có trọng lượng ai,a2, ... ,an

Bài toán xếp Ba lô 0-1 (KNASPACK) cần xác định tập chỉ số

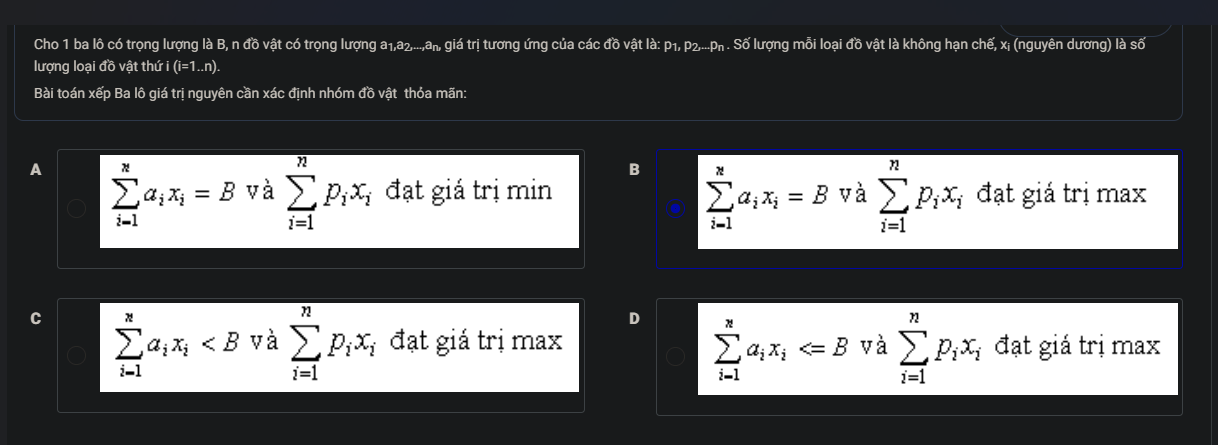
Tc{1,2, ... ,n} sao cho:



**Câu 21: Cho 1 ba lô có trọng lượng là B, n đồ vật có trọng lượng a1,a2, ... ,an, giá trị tương ứng của các đồ vật là:**

p1, P2, ... Pn - Số lượng mỗi loại đồ vật là không hạn chế, xị (nguyên dương) là số lượng loại đồ vật thứ i (i=1 .. n).

Bài toán xếp Ba lô giá trị nguyên cần xác định nhóm đồ vật thỏa mãn:



Ý D nhé

**Câu 22: Bài toán nào sau đây thuộc lớp NP nhưng không phải NPC:**

A

Bài toán xác định chu trình Hamiltonian

B

Bài toán Tối ưu hóa tuyến tính

C

Bài toán tập phủ đỉnh tối ưu

D

Bài toán TSP

**Câu 23: Bài toán 3-SAT thuộc lớp bài toán nào:**

A

NP

B

NPC

C

NP-Hard

D

P

**Câu 24: Bài toán A được gọi là NPC nếu:**

A

A là bài toán quyết định và A không là bài

toan trong NP, mọi bài toan trong NP đều

có thể dẫn về được A

B

A là bài toán quyết định và A là bài toán

trong NP, mọi bài toán trong NP đều có

thể dẫn về được A

C

A là bài toan trong NP, mọi bài toán trong

NP đều có thể dẫn về được A.

D

A là bai toan trong NP, tồn tại bài toán

trong NP dẫn về được A

**câu 25: Lớp NP-Hard bao gồm những bài toán nào:**

A

Những bài toan mà tất ca các bài toán

trong NP có thể quy dẫn đến chúng

B

Những bài toán không thể giải quyết

trong thời gian đa thức

C

Những bài toan có thể giải quyết trong

thời gian đa thức

D

Những bài toán có thể kiểm định nghiệm

trong thời gian đa thức

**câu 26 : Cho A, B, C là các bài toán. Nếu A dẫn về được B và B dẫn về được C thì:**

A

C dẫn về được A

B

A dẫn về được C

C

C dẫn về được B

D

B dẫn về được A

**Câu 27: Bài toán nào sau đây là NPC và được gọi là “ bài toán khó dễ nhất":**

A

Bài toán 3- SAT

B

Bài toán phân hoạch

C

Bài toán phủ đỉnh

D

Bài toan xác định chu trình Hamilton

**Câu 28 : Bài toán phủ đỉnh(Vertex Cover- VC) thuộc lớp bài toán nào:**

A

P

B

NP-Hard

C

NPC

D

NP

**Câu 29 : Bài toán lập lịch thuộc lớp bài toán nào:**

A P

B NPC

C NP

D NP-Hard

**Câu 30 : Nếu bài toán A có thể quy dẫn đến bài toán B, và bài toán B thuộc lớp NPC, điều gì đúng:**

A

Bài toán A không thuộc lớp NP

B

Bài toan A thuộc lớp NPC

C

Bài toán A không thể giải quyết được

D

Bài toán A thuộc lớp NP-Hard

**Bài 6**

**Câu 1: Lược đồ chung của kỹ thuật chia để trị :**

A

void DivideConquer(A,x); (Tìm nghiệm x của bài toán A}

{ Chia A thành các bài toán con A1, A2, ... ,Ami

for (i=1; i <= m; i++) DivideConquer(Aj,Xi);

Kết hợp các nghiệm Xị (i = 1, 2, ... , m) của các bài toán con Ai để nhận được

nghiệm x cua bai toán A;}

B

{

void DivideConquer(A,x); (Tìm nghiệm x của bài toán A}

Chia A thanh cac bai toan con A1, A2, ... , Am

for (i=1; i <= m; i++)

Kết hợp cac bai toan con Ai để nhận được bài toán A; }

C

void DivideConquer(A,x); {Tìm nghiệm x cúa bài toán A}

{ If A đủ nhỏ then Giái bài toán A;

Else

Chia A thanh cac bai toan con A1, A2, ... , Ami

for (i=1; i <= m; i++) DivideConquer(Aj,Xi);

Kết hợp các nghiệm Xị (i = 1, 2, ... , m) của các bài toán con Ai để nhận được

nghiệm x của bài toán A; }}

D

void DivideConquer(A,x); (Tìm nghiệm x của bài toán A}

{ If A đủ nhỏ then Giái bài toán A;

Else

{

Chia A thanh cac bai toan con A1, A2, ... , Am

for (i=1; i <= m; i++) DivideConquer(Aj,Xi);

Kết hợp các bài toán con A¡ để nhận được bài toán A; }}

**Câu2 :** Với thuật toán Quicksort khi sử dụng kỹ thuật “chia để trị" ," quá trình phân chia thể hiện:

A

Phân chia danh sách thành 2 danh sách con “bên trái" và "bên phải", trộn "bên trái" và "bên

phải" để được danh sách có thứ tự

B

Sắp xếp hai danh sách "bên trai" và "bên phai" của khoa chốt để được danh sách không có

thứ tự

C

Phân chia danh sách thành 2 danh sách con "bên trái" và "bên phái", sắp xếp "bên trái" và

"bên phải" để được danh sách có thứ tự

D

Phan chia danh sach thanh 2 danh sach con "bên trai" va "bên phải

**Câu 3 :**

Với bài toán xếp lịch thi đấu thể thao khi sử dụng kỹ thuật “chia để trị", bài toán cơ sở có dạng:

A

Xếp lịch thi đấu cho 4 cầu thủ

B

Xếp lịch thi đấu cho 2 cầu thú

C

Xếp lịch thi đấu cho 1 cầu thủ

D

Xếp lịch thi đấu cho 3 cầu thủ

**Câu 4:** Với bài toán xếp lịch thi đấu thể thao khi sử dụng kỹ thuật “chia để trị", quá trình phân chia thể hiện:

A

Để xếp lịch cho n cầu thủ, ta xếp lịch cho 2n cầu thủ

B

Để xếp lịch cho n cầu thủ, ta xếp lịch cho 2n cầu thủ

C

Để xếp lịch cho n cầu thủ, ta xếp lịch cho n/2 cầu thủ; để xếp lịch cho

n/2 cầu thủ, ta xếp lịch cho n/4 cầu thủ, ...

D

Để xếp lịch cho n cầu thủ, ta xep lịch cho n/4 cầu thủ; để xếp lịch cho

n/4 cầu thủ, ta xếp lịch cho 4 cầu thủ, ...

câu 5 : Với bài toán tìm kiếm nhị phân giá trị x trên một dãy đã sắp xếp, quá trình chia để trị được thể hiện:

A

- Tìm phần tử ở vị trí số 2 của dãy (khóa)

- So sánh x với phần tử khóa

- Nếu bang nhau thì trả về vị trí số 2

- Nếu x nhỏ hơn thì tìm ở nửa bên trái

- Nếu x lớn hơn thì tìm ở nửa bên phải

- Trả về giá trị 0 (nếu không tìm thấy)

B

- Tìm phần tử ở giữa dãy

- So sánh x với phần tử ở giữa dãy

- Nếu bầng nhau thì trả về vị trí giữa

- Nếu x nhỏ hơn thì tìm ở nửa bên trái

- Nếu x lớn hơn thì tìm ở nửa bên phải

- Trả về giá trị 0 (nếu không tìm thấy)

C

- Tìm phần tử ở vị trí số n-1 của dãy (khóa)

- So sánh x với phần tử khóa

- Nếu bang nhau thì trả về vị trí số n-1

- Nếu x nhỏ hơn thì tìm ở nửa bên phải

- Nếu x lớn hơn thì tìm ở nửa bên trái

- Trả về giá trị 0 (nếu không tìm thấy)

D

- Tìm phần tử ở giữa dãy

- So sánh x với phần tử ở giữa dãy

- Nếu bằng nhau thì trả về vị trí giữa

- Nếu x nhỏ hơn thì tìm ở nửa bên phải

- Nếu x lớn hơn thì tìm ở nửa bên trái

- Trả về giá trị 0 (nếu không tìm thấy)

Câu 6: Với bài toán tìm kiếm nhị phân giá trị x trên một dãy đã sắp xếp, bài toán cơ sở có dạng:

A

Tìm kiếm trong một dãy có số phần tử còn một phần tư

B

Tìm kiếm trong một dãy chỉ gồm một phần tử

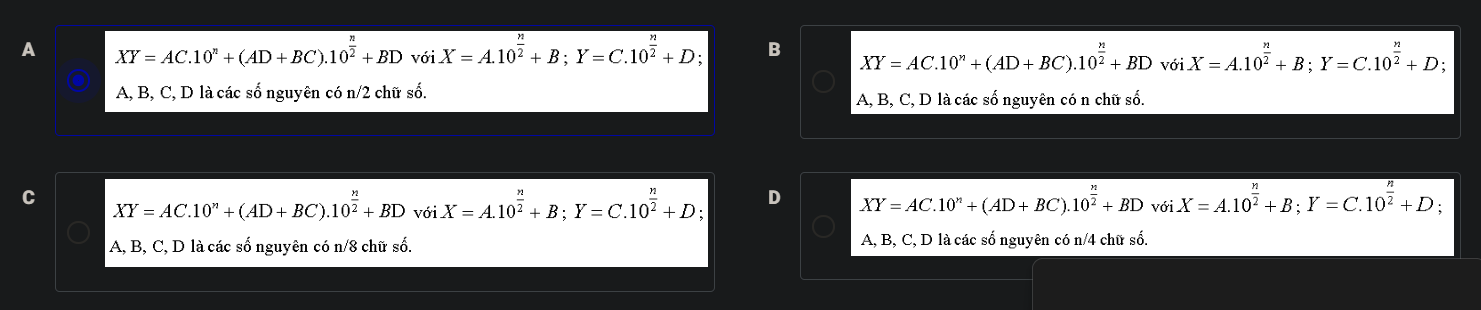
C

Tìm kiếm trong một dãy chỉ gồm hai phần tử

D

Tìm kiếm trong một dãy có số phần tử còn một nửa

Câu 7: Xét bài toán nhân 2 số nguyên lớn có n chữ số X và Y, công thức để tổng hợp kết quả của bài toán khi sử dụng kỹ thuật chia để trị là:



Câu 8: Khi giải quyết bài toán Tháp Hà Nội

"Cho 3 cot A, B, C. Trên cột A đặt n cai đĩa với kích co khac nhau, theo thứ tự to dần đến nhỏ dần từ dưới lên. Hãy di chuyển n cái đĩa từ cột A sang cột C, sao cho:

1. Mỗi bước chỉ có thể chuyển 1 cái đĩa từ cột này sang cột khác, cái đĩa được nhấc ra phải là cái đĩa ở trên cùng (không được đi chuyển cái đĩa khi có đĩa khác ở trên nó)

2. Khi chuyển đĩa sang một cột thì phai đặt nó ở trên cùng.

3. Không được đặt một cái đĩa to len trên cái đĩa nhỏ hơn. Tức là một đĩa chỉ có thể được chuyển vào một cột trống hoặc cột đang có đĩa to hơn nó ở trên cùng."

ý tưởng chia để trị:

A

1.Chuyển n-1 đĩa từ cột A sang cột C;

2.Chuyển 1 đĩa từ cột A sang cột C;

3.Chuyển n-1 đĩa từ cột C sang cột B.

B

1.Chuyển 1 đĩa từ cột A sang cột B;

2.Chuyển n-1 đĩa từ cột A sang cột C;

3.Chuyển n-1 đĩa từ cột B sang cột C.

C

1. Chuyển n-1 đĩa từ cột A sang cột B;

2. Chuyển một đĩa (thứ n) từ cột A sang cột C;

3. Chuyển n-1 đĩa từ cột B sang cột C.

D

1.Chuyển 2 đĩa từ cột A sang cột B;

2.Chuyển n-1 đĩa từ cột A sang cột C;

3.Chuyển n-1 đĩa từ cột B sang cột C.

Câu 9 : Kỹ thuật chia để trị hoạt động bằng cách:

A

Chia nhỏ mot bài toan phức tạp thành cac bài toan con nhỏ hơn, sau đo kết hợp các

bài toán con để có được lời giải cho bài toán ban đầu.

B

Chia nhỏ một bài toán phức tạp thành các bài toán con nhỏ hơn, giải quyết từng bài

toán con một cách độc lập, sau đó tổng hợp các kết quả lại để có được lời giải cho

bài toán ban đầu

C

Chia nhỏ mot bài toan phức tạp thành cac bài toan con độc lập, sau đó kết hợp để có

được lời giải cho bài toán ban đầu.

D

Chia nhỏ một bài toán phức tạp thành các bài toán con nhỏ hơn, giải quyết từng bài

toán con một cách độc lập để có được lời giải cho bài toán ban đầu.

câu 10 : Các bước chính trong kỹ thuật "Chia để trị" gồm:

A

1. Chia đôi;

2. Giải quyết ;

3. Tổng hợp

B

1. Chia nhỏ ;

2. Tổng hợp ;

3. Giải quyết

C

1. Chia nhỏ

2. Giải quyết

3. Tổng hợp

D

1. Chia đôi;

2. Tổng hợp ;

3. Giải quyết

Câu 11 : Bước chia nhỏ trong kỹ thuật chia để trị có ý nghĩa là:

A

Bài toan A ban đầu được chia thanh nhiều bài toán con, mỗi bài toán

con có kích thước lớn hơn bài toán A

B

Bài toán A ban đầu được chia thành nhiều bài toán con nhỏ hơn, độc lập

với nhau, có cau truc tương tự như bai toan ban đầu nhưng với kích

thước nhỏ hơn.

C

Bài toán A ban đầu được chia thành nhiều bài toán con nhỏ hơn, độc lập

với nhau, có cấu trúc tương tự như bài toán ban đầu nhưng với kích

thước lớn hơn.

D

Bai toan A ban đầu được chia thanh nhiều bài toan con, mỗi bài toan

con có kích thước bằng kích thước của bài toán A.

câu 12: Kỹ thuật chia để trị được thiết kế theo kiểu:

A

Từ phải sang trái (right - left)

B

Từ trên xuống (top - down)

C

Từ dưới lên (bottom - up)

D

Từ trái sang phải (left - right)

Câu 13:

Việc tổng hợp lời giải của các bài toán con để nhận được lời giải cho bài toán cần giải quyết trong kỹ thuật “chia để trị” có thể không cần thực hiện, trong trường

hợp:

A

Quá trình phân chia bài toán ban đầu thành các bài toán cơ sở đã chứa

đựng việc tổng hợp kết quả. Khi giải xong các bài toán cơ sở thì bài toán

ban đầu cũng đã được giải quyết.

B

Bài toan ban đầu đa được phân chia hết

C

Các bài toán con nhận được không cần phân chia nữa

D

Các bài toán cơ sở đã được giải hết

Câu 14 : Với thuật toán Mergesort khi sử dụng kỹ thuật “chia để trị” quá trình phân chia thể hiện:

A

Chia đoi một danh sách, cho đến khi danh sách chỉ còn hai phần tử

B

Chia đoi một danh sach, cho đen khi danh sach chỉ còn một nửa số

phần tử

C

Chia đoi mot danh sach, cho den khi danh sach chi con mot phan tu so

phần tử

D

Chia đoi một danh sách, cho đến khi danh sách chỉ còn một phần tử

Câu 15 : Với thuật toán Mergesort khi sử dụng kỹ thuật “chia để trị”, việc tổng hợp kết quả là :

A

“trộn" 2 danh sách chưa có thứ tự để được một danh sách có thứ tự.

B

"trộn" 2 danh sách đã có thứ tự để được một danh sách có thứ tự.

C

"tron" 2 danh sách chua có thứ tự để được một danh sách chưa có thứ

tự.

D

"trộn" 2 danh sách đã có thứ tự để được một danh sách không có thứ

tự.