

Câu 1:

Tìm mô tả đúng nhất cho hàm TinhTong sau:

```
int TinhTong(int N)
{ int so = 2; int tong = 0; int dem = 0;
while (dem <N)
{
if (KiemTra(so) == 1)
{
tong = tong + so;
dem ++;
}
so = so + 1;
}
return tong;
} Trong đó
int KiemTra(int so)
{
for (int i = 2; i<so; i++)
if (so%i == 0)
return 0;
return 1;
}
```

A. Hàm tính tổng N số nguyên đầu tiên

B. Hàm tính tổng N số nguyên tố nhỏ hơn N

C. Cả a, b đều sai

D. Cả a, b đều đúng

Câu 2:

Mối quan hệ giữa cấu trúc dữ liệu và giải thuật có thể minh họa bằng đẳng thức:

A. Cấu trúc dữ liệu + Giải thuật = Chương trình

B. Cấu trúc dữ liệu + Chương trình = Giải thuật

C. Chương trình + Giải thuật = Cấu trúc dữ liệu

D. Cấu trúc dữ liệu = Chương trình

Câu 3:

Các tiêu chuẩn đánh giá cấu trúc dữ liệu. Để đánh giá một cấu trúc dữ liệu chúng ta thường dựa vào một số tiêu chí:

A. Cấu trúc dữ liệu phải tiết kiệm tài nguyên (bộ nhớ trong)

B. Cấu trúc dữ liệu phải phản ánh đúng thực tế của bài toán

C. Cấu trúc dữ liệu phải dễ dàng trong việc thao tác dữ liệu

D. Cả a, b, c đều đúng

Câu 4:

Đoạn mã giả dưới đây mô tả thuật toán gì?

Thuật toán:

B1: $k = 1$

B2: IF $M[k] == X$ AND $k \neq N$

B2.1: $k++$

B2.2: Lặp lại B2

B3: IF $k < N$ Thông báo tìm thấy tại vị trí k

B4: ELSE Không tìm thấy.

B5: Kết thúc

A. Tìm nhị phân phần tử có giá trị X

B. Tìm phần tử nhỏ nhất của mảng M bao gồm N phần tử

C. Tìm tuyến tính phần tử có giá trị X

D. Cả a, b, c đều sai

•

Câu 5:

Cho hàm tìm kiếm tuyến tính như sau:

```
int TimKiem (int M[], int N, int X)
```

```
{ int k = 0;
```

```
M[N] = X;
```

```
while (M[k] != X)
```

```
    k++;
```

```
    if (k < N)
```

```
        return (k);
```

```
    return (-1);
```

```
}
```

Chọn câu đúng nhất:

A. Hàm sẽ trả về 0 nếu không tìm thấy phần tử có giá trị là X

B. Hàm sẽ trả về 1 nếu tìm thấy phần tử có giá trị là X

C. Hàm sẽ trả về -1 nếu không tìm thấy phần tử có giá trị là X

D. Hàm sẽ trả về 1 nếu không tìm thấy phần tử có giá trị là X

•

Câu 6:

Xét thủ tục sau:

```
int TimKiemNP (int M[], int First, int Last, int X)
```

```
{
```

```
if (First > Last)

return (-1);

int Mid = (First + Last)/2;

if (X == M[Mid])

return (Mid);

if (X < M[Mid])

return(TimKiemNP (M, First, Mid – 1, X));

else

return(TimKiemNP (M, Mid + 1, Last, X));

}
```

Lựa chọn câu đúng nhất để mô tả thủ tục trên:

- A. Thủ tục hỗ trợ tìm kiếm phân tử có giá trị là X trên mảng các phần tử từ chỉ số từ First đến chỉ số Last
- B. Thủ tục hỗ trợ tìm kiếm đệ quy phân tử có giá trị là X trên mảng các phần tử từ chỉ số từ First đến chỉ số Last
- C. Thủ tục hỗ trợ tìm kiếm đệ quy phân tử có giá trị là X trên mảng các phần tử từ chỉ số từ Last đến chỉ số First
- D. Thủ tục hỗ trợ tìm kiếm không đệ quy phân tử có giá trị là X trên mảng các phần tử từ chỉ số từ Last đến chỉ số First

Câu 7:

Chọn câu đúng nhất để mô tả thuật toán sắp xếp nổi bọt (Bubble Sort) trên mảng M có N phần tử:

A. Đi từ cuối mảng về đầu mảng, trong quá trình đi nếu phần tử ở dưới (đứng phía sau) nhỏ hơn phần tử đứng ngay trên (trước) nó thì hai phần tử này sẽ được đổi chỗ cho nhau. Sau mỗi lần đi chúng ta đưa được một phần tử trôi lên đúng chỗ. Sau N-1 lần đi thì tất cả các phần tử trong mảng M sẽ có thứ tự tăng

B. Đi từ đầu mảng về cuối mảng, trong quá trình đi nếu phần tử ở dưới (đứng phía sau) nhỏ hơn phần tử đứng ngay trên (trước) nó thì hai phần tử này sẽ được đổi chỗ cho nhau. Sau mỗi lần đi chúng ta đưa được một phần tử trôi lên đúng chỗ. Sau N lần đi thì tất cả các phần tử trong mảng M sẽ có thứ tự tăng.

C. Đi từ cuối mảng về đầu mảng, trong quá trình đi nếu phần tử ở dưới (đứng phía sau) nhỏ hơn phần tử đứng ngay trên (trước) nó thì hai phần tử này sẽ được đổi chỗ cho nhau. Sau mỗi lần đi chúng ta đưa được một phần tử trôi lên đúng chỗ. Sau N lần đi thì tất cả các phần tử trong mảng M sẽ có thứ tự tăng

D. Cả a, b, c đều sai

•

Câu 8:

Hàm mô tả sắp xếp nổi bọt (Bubble Sort) trên mảng M có N phần tử

```
void BubbleSort(int M[], int N)
```

```
{
```

```
[2] int Temp;  
[3] for (int I = 0; I < N-1; I++)  
[4] .....  
[5] if (M[J] < M[J-1])  
[6] {  
[7] Temp = M[J];  
[8] M[J] = M[J-1];  
[9] M[J-1] = Temp;  
[10] }  
[11] return;  
[12] }  
[13]
```

Lệnh nào sau đây sẽ được đưa vào dòng lệnh thứ [5] của thủ tục:

A. for (int J = N-1; J > I; J++)

B. for (int J = N; J < I; J--)

C. for (int J = N-1; J > I; J--)

D. Không có dòng lệnh nào phù hợp, không cần thêm vào thuật toán vẫn chạy đúng

-

Câu 9:

Thủ tục mô tả thuật toán sắp xếp chọn trực tiếp (Straight Selection Sort):

```
void SapXepChonTrucTiep(T M[], int N)
```

```
{
```

```
    int K = 0, PosMin;
```

```
    int Temp;
```

```
    while (K < N-1)
```

```
    { T Min = M[K];
```

```
      PosMin = K;
```

```
      for (int Pos = K+1; Pos < N; Pos++)
```

```
      if (Min > M[Pos])
```

```
      {
```

```
        Min = M[Pos];
```

```
        PosMin = Pos
```

```
      }
```

```
      } .....
```

```
      [1] .....
```

```
      [2] .....
```

```
      [3] K++;
```

```
    }
```

```
    return;
```


}

Chọn câu lệnh thích hợp để đưa vào [1], [2], [3] với mục tiêu hoán vị M[K] và M[PosMin]

A. Temp = M[K] ; Temp = M[PosMin]; M[PosMin] = Temp;

B. M[K] = Temp; M[K] = M[PosMin]; M[PosMin] = Temp ;

C. Temp = M[K] ; M[PosMin] = M[K]; M[PosMin] = Temp ;

D. Temp = M[K] ; M[K] = M[PosMin]; M[PosMin] = Temp ;

•

Câu 10:

Đối với thuật toán sắp xếp chọn trực tiếp cho dãy các phần tử sau (10 pt) 16 60 2 25 15 45 5 30 33 20

Cần thực hiện chọn lựa phần tử nhỏ nhất để sắp xếp mảng M có thứ tự tăng dần.

A. 7 lần

B. 8 lần

C. 9 lần

D. 10 lần

•

Câu 11:

Thuật toán sắp xếp chèn trực tiếp (Straight Insertion Sort) được mô tả bằng đoạn mã giả như sau:

B1: $K = 1$

B2: IF ($K = N$) Thực hiện BKT

B3: $X = M[K+1]$

B4: $Pos = 1$

B5: IF ($Pos > K$) Thực hiện B7

B6: ELSE // Tìm vị trí chèn

B6.1: If ($X \leq M[Pos]$) Thực hiện B7

B6.2: $Pos++$

B6.3: Lặp lại B6.1

B7: $I = K+1$ B8: IF ($I > Pos$)

B8.1: $M[I] = M[I-1]$

B8.2: $I--$

B8.3: Lặp lại B8

B9: ELSE

B9.1: $M[Pos] = X$

B9.2: $K++$

B9.3: Lặp lại B2

BKT: Kết thúc Trong đó B8 mô tả trường hợp

A. Nếu còn phải dời các phần tử từ $Pos \rightarrow I$ về phía sau 1 vị trí

B. Nếu còn phải dời các phần tử từ $Pos \rightarrow K+1$ về phía sau 1 vị trí

C. Nếu còn phải dời các phần tử từ $Pos \rightarrow K$ về phía sau 1 vị trí

D. Nếu còn phải dời các phần tử từ $Pos \rightarrow I+1$ về phía sau 1 vị trí

•

Câu 12:

Giả sử cần sắp xếp mảng M có N phần tử sau theo phương pháp sắp xếp chèn trực tiếp 11 16 12
75 51 54 5 73 36 52 98

Cần thực hiện chèn các phần tử vào dãy con đã có thứ tự tăng đứng đầu dãy M để
sắp xếp mảng M có thứ tự tăng dần.

A. 7 lần

B. 8 lần

C. 9 lần

D. 10 lần

•

Câu 13:

Lựa chọn định nghĩa về danh sách đúng nhất?

A. Danh sách là tập hợp các phần tử có kiểu dữ liệu xác định và giữa chúng có một mối liên hệ nào đó

B. Số phần tử của danh sách gọi là chiều dài của danh sách

C. Một danh sách có chiều dài bằng 0 là một danh sách rỗng

D. Cả a, b, c đều đúng

-

Câu 14:

Tìm mô tả đúng cho hàm sau:

```
int SC (int M[], int Len, int CM[])
```

```
{ for (int i = 0; i < Len; i++)
```

```
CM[i] = M[i];
```

```
return (Len);
```

```
}
```

A. Hàm thực hiện việc sao chép nội dung mảng CM có chiều dài Len về mảng M có cùng chiều dài. Hàm trả về chiều dài của mảng M sau khi sao chép

B. Hàm thực hiện việc sao chép nội dung mảng M có chiều dài Len -1 về mảng CM có cùng chiều dài. Hàm trả về chiều dài của mảng CM sau khi sao chép

C. Hàm thực hiện việc sao chép nội dung mảng CM có chiều dài Len -1 về mảng M có cùng chiều dài. Hàm trả về chiều dài của mảng M sau khi sao chép

D. Hàm thực hiện việc sao chép nội dung mảng M có chiều dài Len về mảng CM có cùng chiều dài. Hàm trả về chiều dài của mảng CM sau khi sao chép

-

Câu 15:

Cấu trúc dữ liệu mảng có các ưu điểm nào?

A. Việc thêm, bớt các phần tử trong danh sách đặc có nhiều khó khăn do phải di dời các phần tử khác đi qua chỗ khác

B. Việc truy xuất và tìm kiếm các phần tử của mảng là dễ dàng vì các phần tử đứng liền nhau nên chúng ta chỉ cần sử dụng chỉ số để định vị vị trí các phần tử trong danh sách (định vị địa chỉ các phần tử)

C. Mật độ sử dụng bộ nhớ của mảng là tối ưu tuyệt đối

D. Câu a, b, c đúng

•

Câu 16:

Định nghĩa nào là đúng với danh sách liên kết?

A. Danh sách liên kết là cấu trúc dữ liệu dạng cây

B. Danh sách liên kết là cấu trúc dữ liệu tự định nghĩa

C. Danh sách liên kết là tập hợp các phần tử mà giữa chúng có một sự nối kết với nhau thông qua vùng liên kết của chúng

D. Danh sách liên kết là tập hợp các phần tử mà đặt kề cận với nhau trong vùng nhớ

•

Câu 17:

Định nghĩa cấu trúc dữ liệu của danh sách liên kết đơn được mô tả như sau:

```
typedef struct Node
```

```
{ int Key;
```

```
Node * NextNode;
```

```
} OneNode;
```

Trong đó, khai báo `Node * NextNode;` dùng để mô tả:

A. Con trỏ trỏ tới phần dữ liệu

B. Vùng liên kết quản lý địa chỉ phần tử kế tiếp

C. Con trỏ trỏ tới địa chỉ vùng nhớ của phần tử trước đó trong danh sách liên kết đơn

D. Con trỏ trỏ tới địa chỉ vùng nhớ của phần tử đầu tiên trong danh sách liên kết đơn

-

Câu 18:

Với cấu trúc dữ liệu của danh sách liên kết đơn lưu trữ thông tin về phòng máy:

```
typedef struct PM
```

```
{
```

```
int maPM; int tongsoMay;
```

```
} PHONGMAY;
```

```
typedef struct Node { PHONGMAY Data; Node * NextNode;
```

```
} OneNode;
```

```
typedef OneNode * SLLPointer;
```

Để quản lý danh sách liên kết đơn bằng phần tử đầu và phần tử cuối, cần định nghĩa kiểu dữ liệu:

A. SLLPointer DanhSach;

B. typedef struct SSLLIST { SLLPointer First; SLLPointer Last; } LIST; LIST DanhSach;

C. typedef struct SSLLIST { SLLPointer First; SLLPointer Last; int total; } LIST; LIST DanhSach;

D. typedef struct SSLLIST { SLLPointer First; int total; } LIST; LIST DanhSach;

•

Câu 19:

Tổ chức cấu trúc dữ liệu cho danh sách liên kết đơn:

```
typedef struct Node
```

```
{ int Data; Node * Link;
```

```
} OneNode; typedef OneNode * SLLPointer;
```

Mã giả thuật toán thêm một phần tử có giá trị thành phần dữ liệu là NewData vào trong danh sách liên kết đơn SLList vào ngay sau nút có địa chỉ InsNode:

B1: NewNode = new OneNode

B2: IF (NewNode = NULL) Thực hiện BKT

B3: NewNode ->Link = NULL

B4: NewNode ->Data = NewData

B5: IF (InsNode-> Link = NULL)

B5.1: InsNode-> Link = NewNode

B5.2: Thực hiện BKT // Nối các nút kế sau InsNode vào sau NewNode

B6:

// Chuyển mỗi liên kết giữa InsNode với nút kế của nó về NewNode

B7:

BKT: Kết thúc

B6 và B7 dùng để nối nút kế sau InsNode vào sau NewNode và chuyển mỗi liên kết giữa InsNode với nút kế nó về NewNode.

Hãy chọn câu đúng nhất cho B6 và B7

A. B6: InsNode-> Link = NewNode-> Link B7: NewNode = InsNode-> Link

B. B6: InsNode-> Link = NewNode-> Link B7: InsNode-> Link = NewNode

C. B6: NewNode-> Link = InsNode-> Link B7: NewNode = InsNode-> Link

D. B6: NewNode-> Link = InsNode-> Link B7: InsNode-> Link = NewNode

•

Câu 20:

Với định nghĩa cấu trúc dữ liệu cho danh sách liên kết đơn:

typedef struct Node

{

int Data; Node * Link;

} OneNode;

typedef OneNode * SLLPointer;

Hàm dưới đây để thêm một phần tử có giá trị thành phần dữ liệu là NewData vào trong danh sách liên kết đơn SLList vào ngay sau nút có địa chỉ InsNode.

SLLPointer ThemGiua(SLLPointer &SList, int NewData, SLLPointer &InsNode)

{ SLLPointer NewNode = new OneNode;

if (NewNode != NULL)

NewNode ->NextNode = NULL;

NewNode ->Data = NewData;

else

return (NULL);

if (InsNode->Link == NULL)

{

InsNode-> Link = NewNode; return (SList);

}

.....

.....

return (SList);

}

Hãy lựa chọn câu đúng nhất:

A. InsNode -> Link = NewNode -> Link; InsNode-> Link = NewNode;

B. NewNode-> Link = InsNode-> Link; InsNode-> Link = NewNode;

C. InsNode -> Link = NewNode -> Link; NewNode = InsNode-> Link;

D. NewNode-> Link = InsNode-> Link; NewNode = InsNode-> Link;

•

Câu 21:

Khi lưu trữ cây nhị phân dưới dạng mảng, nếu vị trí của nút cha trong mảng là 3 thì vị trí tương ứng của nút con phải sẽ bao nhiêu trong các phương án sau?

A. 2

B. 4

C. 6

D. 7

Câu 22:

Giải thuật đệ quy là:

A. Trong giải thuật của nó có lời gọi tới chính nó

B. Trong giải thuật của nó có lời gọi tới chính nó nhưng với phạm vi lớn hơn

C. Trong giải thuật của nó có lời gọi tới chính nó nhưng với phạm vi nhỏ hơn

D. Trong giải thuật của nó có lời gọi tới một giải thuật khác đã biết kết quả

-

Câu 23:

Đặc điểm của giải thuật đệ quy:

A. Có một trường hợp đặc biệt, trường hợp suy biến Khi trường hợp này xảy ra thì bài toán còn lại sẽ được giải quyết theo một cách khác

B. Trong thủ tục đệ quy có lời gọi đến chính thủ tục đó

C. Sau mỗi lần có lời gọi đệ quy thì kích thước của bài toán được thu nhỏ hơn trước

D. Tất cả đều đúng

-

Câu 24:

Danh sách tuyến tính là:

A. Danh sách dạng được lưu dưới dạng mảng

B. Danh sách tuyến tính là một danh sách rỗng

C. Danh sách mà quan hệ lân cận giữa các phần tử được xác định

D. Danh sách tuyến tính là một danh sách có dạng (a_1, a_2, \dots, a_n)

•

Câu 25:

Ưu điểm của việc cài đặt danh sách bằng mảng:

A. Có thể thay đổi số lượng phần tử theo ý muốn của người dùng

B. Có thể bổ sung hoặc xóa một phần tử bất kỳ trong mảng

C. Việc truy nhập vào phần tử của mảng được thực hiện trực tiếp dựa vào địa chỉ tính được (chỉ số), nên tốc độ nhanh và đồng đều đối với mọi phần tử

D. Tất cả các ý trên đều đúng

•

Câu 26:

Danh sách tuyến tính dạng ngăn xếp là:

A. Là một danh sách tuyến tính trong đó phép bổ sung một phần tử vào ngăn xếp được thực hiện ở một đầu, Và phép loại bỏ không thực hiện được

B. Là một danh sách tuyến tính trong đó phép bổ sung một phần tử vào ngăn xếp được thực hiện ở một đầu , và phép loại bỏ được thực hiện ở đầu kia

C. Là một danh sách tuyến tính trong đó phép bổ sung một phần tử vào ngăn xếp và phép loại bỏ một phần tử khỏi ngăn xếp luôn luôn thực hiện ở tại một vị trí bất kì trong danh sách

D. Là một danh sách tuyến tính trong đó phép bổ sung một phần tử vào ngăn xếp và phép loại bỏ một phần tử khỏi ngăn xếp luôn luôn thực hiện ở một đầu gọi là đỉnh

•

Câu 27:

Định nghĩa danh sách tuyến tính Hàng đợi (Queue):

A. Hàng đợi là kiểu danh sách tuyến tính trong đó, phép bổ sung một phần tử được thực hiện ở một đầu, gọi là lối sau (rear) hay lối trước (front). Phép loại bỏ không thực hiện được

B. Là một danh sách tuyến tính trong đó phép bổ sung một phần tử và phép loại bỏ một phần tử được thực hiện ở tại một vị trí bất kì trong danh sách

C. Hàng đợi là kiểu danh sách tuyến tính trong đó, phép bổ sung phần tử ở một đầu, gọi là lối sau (rear) và phép loại bỏ phần tử được thực hiện ở đầu kia, gọi là lối trước (front)

D. Hàng đợi là kiểu danh sách tuyến tính trong đó, phép bổ sung một phần tử hay loại bỏ được thực hiện ở một đầu danh sách gọi là đỉnh (Top)

-

Câu 28:

Hàng đợi còn được gọi là danh sách kiểu:

A. LOLO

B. FIFO

C. FILO

D. LIFO

-

Câu 29:

Cho dãy số {6 1 3 0 5 7 9 2 8 4}. áp dụng phương pháp sắp xếp lựa chọn (Select sort) sau lần lặp đầu tiên của giải thuật ta có kết quả: {0 1 3 6 5 7 9 2 8 4}. Dãy số thu được sau lần lặp thứ hai là:

A. {0 1 2 6 5 7 9 3 8 4}

B. {0 1 3 6 5 7 9 2 8 4}

C. {0 1 2 3 4 5 6 7 8 9}

D. {0 1 2 6 5 7 9 3 4 8}

•

Câu 30:

Cho dãy số {6 1 3 0 5 7 9 2 8 4}. áp dụng phương pháp sắp xếp lựa chọn (Select sort) sau lần lặp đầu tiên của giải thuật ta có kết quả: {0 1 3 6 5 7 9 2 8 4}. Dãy số thu được sau lần lặp thứ ba là:

A. {0 1 2 6 5 7 9 3 8 4}

B. {0 1 2 6 5 7 9 3 4 8}

C. {0 1 2 3 6 5 7 9 8 4}

D. {0 1 2 3 4 5 6 7 8 9}

•

Câu 31:

Cho dãy số {6 1 3 0 5 7 9 2 8 4}. áp dụng phương pháp sắp xếp lựa chọn (Select sort) sau lần lặp đầu tiên của giải thuật ta có kết quả: {0 1 3 6 5 7 9 2 8 4}. Dãy số thu được sau lần lặp thứ tư là:

A. {0 1 2 3 6 5 7 9 8 4}

B. {0 1 2 3 4 5 6 7 8 9}

C. {0 1 2 3 5 7 9 4 8 6}

D. {0 1 2 3 5 7 9 6 8 4}

•

Câu 32:

Cho dãy số {6 1 3 0 5 7 9 2 8 4}. áp dụng phương pháp sắp xếp lựa chọn (Select sort) sau lần lặp đầu tiên của giải thuật ta có kết quả: {0 1 3 6 5 7 9 2 8 4}. Dãy số thu được sau lần lặp thứ năm là:

A. {0 1 2 3 6 5 7 9 8 4}

B. {0 1 2 3 5 7 9 4 8 6}

C. {0 1 2 3 4 5 6 7 8 9}

D. {0 1 2 3 4 7 9 6 8 5}

•

Câu33:

Cho dãy số {6 1 3 0 5 7 9 2 8 4}. áp dụng phương pháp sắp xếp lựa chọn (Select sort) sau lần lặp đầu tiên của giải thuật ta có kết quả: {0 1 3 6 5 7 9 2 8 4}. Dãy số thu được sau lần lặp thứ sáu là:

A. {0 1 2 3 4 7 9 6 8 5}

B. {0 1 2 3 4 5 6 9 8 7}

C. {0 1 2 3 4 5 6 7 8 9}

D. {0 1 2 3 4 5 9 6 8 7}

-

Câu 34:

Cho dãy số $\{3\ 1\ 6\ 0\ 5\ 4\ 8\ 2\ 9\ 7\}$. áp dụng phương pháp sắp xếp nhanh (Quick sort) sau lần lặp đầu tiên của giải thuật ta có kết quả: $\{(0\ 1\ 2)\ 3\ (5\ 4\ 8\ 6\ 9\ 7)\}$. Dãy số thu được sau lần lặp thứ bốn là:

A. $\{(0)\ 1\ (2\ 3)\ 4\ (5\ 6)\ 7\ (8\ 9)\}$

B. $\{0\ 1\ 2\ 3\ (5\ 4\ 8\ 6\ 9\ 7)\}$

C. $\{(3)\ 1\ (6\ 0)\ 5\ (4\ 8)\ 2\ (9\ 7)\}$

D. $\{0\ 1\ (2)\ 3\ (5\ 4)\ 8\ (6\ 9\ 7)\}$

-

Câu 35:

Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. Áp dụng phương pháp sắp xếp lựa chọn, sau lượt 1 dãy sẽ được sắp xếp lại như thế nào?

A. 15 25 40 75 30 55 65 90 85 95

B. 40 25 75 15 30 65 55 90 85 95

C. 15 25 75 40 65 55 90 30 95 85

D. 15 40 25 30 75 65 55 85 90 95

•

Câu 36:

Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. Áp dụng phương pháp sắp xếp lựa chọn, sau lượt 2 dãy sẽ được sắp xếp lại như thế nào?

A. 15 25 75 30 40 65 55 90 85 95

B. 15 40 25 75 30 55 65 90 85 95

C. 15 25 75 40 65 55 90 30 95 85

D. 15 40 25 30 75 65 55 85 90 95

•

Câu 37:

Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. Áp dụng phương pháp sắp xếp hòa nhập (Merge_Sort) trực tiếp, sau lượt 1 dãy sẽ được sắp xếp lại như thế nào?

A. [15 40] [30 25] [55 65] [75 85] [90 95]

B. [40 25] [55 15] [30 65] [75 90] [85 95]

C. [15 25] [40 75] [30 55] [65 90] [85 95]

D. [25 40] [15 75] [55 65] [30 90] [85 95]

•

Câu 38:

Cho dãy số sau: 14 32 10 43 57 87 55 36 97 11. Áp dụng phương pháp tìm kiếm tuần tự, sau bao nhiêu lần thực hiện phép so sánh ta sẽ tìm thấy số 43?

A. 2 lần

B. 3 lần

C. 4 lần

D. 5 lần`

•

Câu 39:

Cho dãy số sau: 10 11 14 32 36 43 55 57 87 97 . Áp dụng phương pháp tìm kiếm nhị phân, sau bao nhiêu lần phân đoạn ta sẽ tìm thấy số 43?

A. 2 lần

B. 3 lần

C. 4 lần

D. 5 lần

•

Câu 40:

Cho dãy số sau: 10 11 14 32 36 43 55 57 87 97. Áp dụng phương pháp tìm kiếm nhị phân, để tìm kiếm số 10, lần phân đoạn thứ nhất của dãy sẽ là:

A. [14 32 10 43 57]

B. [10 11 14 32 36]

C. [87 55 36 97 11]

D. [55 36 97 11]