CÂU HỎI ÔN TẬP MÔN
CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

1/ Giải thuật đệ quy là:

Trong giải thuật của nó có lời gọi tới chính nó nhưng với phạm v i(kích thước) nhỏ hơn

2/ Cho hàm đệ qui sau:

```
Function  Factorial(n)
Begin
  if n= 0 then Factorial:=1 _
  else    Factorial := n*Factorial(n-1);
End;
```

Sau mỗi lần gọi đệ quy thì giá trị của n là

Giảm đi 1 đơn vị

3/ Cho hàm đệ qui sau:

```
Function  Factorial(n)
Begin
  if n=0 then Factorial:=1
  else    Factorial := n*Factorial(n-1);
End;
```

Dòng lệnh if n=0 then Factorial:=1 là:

Điều kiện dừng đệ quy

4/ Hàm đệ qui sau giải bài toán gì?

```
Function  Factorial(n)
Begin
  if n=0 then Factorial:=1
  else    Factorial := n*Factorial(n-1);
End;
```

Tính giai thừa n (tính giá trị của n!)

5/ Cho hàm đệ qui sau:

```
Function  Factorial(n)
Begin
        if n=0 then Factorial:=1
        else    Factorial := n*Factorial(n-1); _
End;
```

Kết quả bằng bao nhiêu khi n=3

6

6/ Hàm đệ qui cho kết quả thế nào khi n=4?

```
Function  Factorial(n)
Begin
        Factorial := n*Factorial(n-1);
End;
```

Lặp vô hạn vì không có điều kiện dừng

7/ Dãy số Fibonacci bắt nguồn từ bài toán cổ về việc sinh sản của các cặp thỏ. Bài toán được đặt ra như sau:
Các con thỏ không bao giờ chết.
Hai tháng sau khi ra đời một cặp thỏ mới sẽ sinh ra một cặp thỏ con.
Khi đã sinh con rồi thì cứ mỗi tháng tiếp theo chúng lại sinh được một cặp con mới.
Giả sử bắt đầu từ một cặp thỏ mới ra đời thì đến tháng thứ 5 sẽ có bao nhiêu cặp?

5

8/ Cho giải thuật đệ quy sau:

```
Function F(n)
Begin
  if n<=2 then   F:=1
  else F := F(n-1) + F(n-2);
End;
```

Dòng lệnh if n<=2 then F:=1 đóng vai trò:

Điều kiện dừng đệ quy

9/ Cho giải thuật đệ quy sau:

```
Function F(n:integer):integer;
Begin
  if n<=2 then F:=1
  else F:= F(n-2) + F(n-1)
End;
```

Khi n=4 kết quả của bài toán trên là:

3

10/ Đặc điểm của giải thuật đệ quy:

Trong thủ tục đệ quy có lời gọi đến chính thủ tục đó.
Sau mỗi lần có lời gọi đệ quy thì kích thước của bài toán được thu nhỏ hơn trước."
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.

11/ giải thuật đệ quy của bài toán Tháp Hà Nội như sau:

```
Procedure Chuyen(n, A, B, C)
Begin
        if n=1 then chuyển đĩa từ A sang C
        else begin
                call Chuyen(n-1, A, C, B);
                call Chuyen(1, A, B, C);
                call Chuyen(n-1, B, A, C) ;
            end;
End;
```

Khi n=3 có bao nhiêu bước chuyển?

7

12/ Danh sách tuyến tính là:

- *Danh sách (List) là tập hợp hữu hạn các phần tử có kiểu dữ liệu xác định, có trình tự sắp xếp thứ tự: đứng trước, đứng sau, đứng giữa các phần tử. (Danh sách mà quan hệ lân cận giữa các phần tử lân cận được xác định)*

13/ ưu điểm của việc cài đặt danh sách bằng mảng:

- *Cài đặt đơn giản.Các phép toán được triển khai dễ dàng.*

14/ Danh sách tuyến tính dạng ngăn xếp là:

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.

15/ Danh sách tuyến tính dạng ngăn xếp làm việc theo nguyên tắc:

LIFO(last in first out): vào sau, ra trước

16/ Khi đổi một số nguyên từ hệ thập phân sang hệ nhị phân thì người ta dùng phép chia liên tiếp cho 2 và lấy các số dư (là các chữ số nhị phân) theo chiều ngược lại.
Cơ chế sắp xếp này chính là cơ chế hoạt động của cấu trúc dữ liệu:

Stack (ngăn xếp)

17/ S là ngăn xếp , Phép toán thêm phần tử vào ngăn xếp là Push, phép lấy ra một phần tử từ ngăn xếp là POP, thủ tục sau làm nhiệm vụ gì?

```
Procedure Chuyen_doi(N);
  While N <> 0 do
   R := N mod 2; {tính số dư trong phép chia N cho 2}
   call PUSH(S, R);
   N := N div 2; {thay N bằng thương của phép chia N cho 2}
end; &
While not Empty(S) do
begin
   call POP(S, R);
   write(R);
end
end.
```

ứng dụng ngăn xếp để đổi số N từ cơ số 10 sang cơ số 2

18/ định nghĩa danh sách tuyến tính Hàng đợi (Queue)

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).

19/ Hàng đợi còn được gọi là danh sách kiểu:

FIFO (vào trước ra trước)

20/ Để thêm một đối tượng x bất kỳ vào Stack, thao tác thường dùng là:

Push(x)

21/ Để loại bỏ một đối tượng ra khỏi Stack, thao tác thường dùng là:

Pop(x)

22/ Để biểu diễn Stack, ta thường sử dụng kiểu dữ liệu nào sau đây:

Danh sách móc nối, mảng dữ liệu

23/ Thao tác POP(x) dùng trong Stack là để:

Loại bỏ một đối tượng x ra khỏi Stack

24/ Thao tác Push(x) dùng trong Stack là để:

Thêm một đối tượng x vào Stack

25/ Cho Stack gồm 5 phần tử {12, 5, 20, 23, 25}, trong đó 25 là phần tử ở đỉnh Stack. Để lấy ra phần tử thứ 4 trong Stack ta phải làm thế nào?

Pop(25), pop(23), push(25)

26/ Cho Stack gồm 5 phần tử {12, 5, 20, 23, 25}, trong đó 25 là phần tử ở đỉnh Stack. Để lấy ra phần tử thứ 5 trong Stack ta phải làm thế nào?

Pop(25), pop(23), push(23)

27/ Cho Stack gồm 5 phần tử {12, 5, 20, 23, 25}, trong đó 25 là phần tử ở đỉnh Stack. Để lấy ra phần tử thứ 3 trong Stack ta phải làm thế nào?

Pop(25), pop(23), pop(20), push(23), push(25)

28/ Trong lưu trữ dữ liệu kiểu Stack, giải thuật sau thực hiện công việc gì?

```
Procedure F(X)
    Begin
        T:=T+1;
        S[T]:=X;
    End;
```

Bổ sung một phần tử vào Stack

29/ Trong lưu trữ dữ liệu kiểu Stack, giải thuật F chính là:

```
Procedure F(X)
    Begin
        T:=T+1;
        S[T]:=X;
    End;
```

Push()

30/ Trong lưu trữ dữ liệu kiểu Stack, giải thuật sau thực hiện công việc gì?

```
Function P
    Begin
                    T:=T-1;
                    P:=S[T+1];
    End;
```

Loại bỏ một phần tử ra khỏi Stack.

31/ Trong lưu trữ dữ liệu kiểu Stack, giải thuật P chính là:

```
Function P
    Begin
        T:=T-1;
        P:=S[t+1];
    End;
```

Pop()

32/ Với đoạn mã sau, nếu n=13, trong Stack sẽ là:

```
While n<>0 do
        begin
            R:=n mod 2;
            Push(R);
            n:=n div 2;
        end;
```

1101

33/ Với đoạn mã sau, nếu n=13, trong các phần tử được bổ sung vào Stack theo thứ tự:

```
While n<>0 do
        begin
            R:=n mod 2;
           Push(R);
           n:=n div 2;
        end;
```

"1 , 0 , 1, 1"

34/ Với đoạn mã sau, nếu các phần tử được đưa vào Stack theo thứ tự : 1 1 0 1 thì các phần tử được loại khỏi Stack theo thứ tự nào?

```
While T>0 do
        begin
            R:=POP(S[T]);
            write(R);
        end;
```

1 0 1 1

35/ Trong lưu trữ dữ liệu kiểu Queue (Q) dưới dạng mảng nối vòng, giả sử F là con trỏ trỏ tới lối trước của Q, R là con trỏ trỏ tới lối sau của Q. Điều kiện F=R=0 nghĩa là:

Queue rỗng

36/ Trong lưu trữ dữ liệu kiểu Queue (Q), giả sử F là con trỏ trỏ tới lối trước của Q, R là con trỏ trỏ tới lối sau của Q. Khi thêm một phần tử vào Queue, thì R và F thay đổi thế nào?

"F không thay đổi, R=R+1."

37/ Trong lưu trữ dữ liệu kiểu Queue (Q), giả sử F là con trỏ trỏ tới lối trước của Q, R là con trỏ trỏ tới lối sau của Q. Khi loại bỏ một phần tử khỏi Queue, thì R và F thay đổi thế nào?

"F=F+1, R không thay đổi."

38/ Giải thuật sau thực hiện việc gì?

```
Procedure Q(x)
Begin
    if R=n then R:=1 else R:=R+1;
    if F=R then begin   write('full')
                        return
                end ;
    Q[R]:=X;
    if F=0 then F:=1;
End;
```

Bổ sung một phần tử vào Queue

39/ Giải thuật sau thực hiện việc gì?

```
Function Q:kiểu dữ liệu;
Begin
   if F=0 then begin   write('NULL')
                          return
                end;
   Y:=Q[F];
   if F=R then begin
                     F:=R:=0;
                     return
                end;
  if F=n then F:=1
                else F:=F+1;
 Q:=Y;
End;
```

Loại bỏ một phần tử ra khỏi Queue

40/ Giải thuật sau thực hiện việc gì?

```
Function P(l:ds): boolean;
        Begin
                P:= (l.last =0);
        End;
```

"Kiểm tra danh sách có rỗng hay không."

41/ Giải thuật sau thực hiện việc gì?

```
 Procedure  P( l:ds);
        Begin
                l.last := 0;
        End;
```

Làm rỗng danh sách. (khởi tạo danh sách rỗng)

42/ Giải thuật sau thực hiện việc gì?

```
  Procedure  F(x,P: integer);
        Begin
                for i:= (l.last +1)  downto (P+1)  do
                                        l.s[i]:=l.s[i-1];
                l.s[P]:=x;
                l.last:=l.last + 1;
        End;
```

"Chèn phần tử x vào vị trí P trong danh sách."

43/ Giải thuật sau thực hiện việc gì?

```
Procedure  F(P: integer);
        Begin
                for i:= P to (l.last -1) do
                                l.s[i]:=l.s[i+1];
                l.last:=l.last -1;
```

End;
"Xoá một phần tử tại vị trí P trong danh sách."

44/ Trong biểu diễn dữ liệu dưới dạng cây, cấp của cây:
Là cấp cao nhất của một nút trên cây.

45/ Trong biểu diễn dữ liệu dưới dạng cây, nút có cấp bằng 0 gọi là:
Nút lá

46/ " Mỗi nút trong cây có tối đa:"
Căn cứ vào hình

47/ " Mỗi nút trong cây có tối đa:"
Căn cứ vào hình

48/ Khi lưu trữ cây nhị phân dưới dạng mảng, nếu vị trí của nút cha trong mảng là i thì vị trí của nút con trái là:
2i

49/ Khi lưu trữ cây nhị phân dưới dạng mảng, nếu vị trí của nút cha trong mảng là i thì vị tí của nút con phải là:
2i+1

50/ 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 sẽ là:
6 và 7

51/ 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 trái sẽ là:
6

52/ 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 là:
7

53/ Duyệt cây nhị phân theo thứ tự trước được thực hiện theo thứ tự:
Gốc – trái – phải (+ Thăm gốc
+ Duyệt cây con trái theo thứ tự trước
+ Duyệt cây con phải theo thứ tự trước)

54/ Duyệt cây nhị phân theo thứ tự giữa được thực hiện theo thứ tự:
Trái – gốc – phải (+ DuyÖt c©y con tr¸i theo thø tù gi÷a
+ Th¨m gèc
+ DuyÖt c©y con ph¶i theo thø tù gi÷a)

55/ Duyệt cây nhị phân theo thứ tự sau được thực hiện theo thứ tự:
Trái – phải – gốc (+ DuyÖt c©y con tr¸i theo thø tù sau
+ DuyÖt c©y con tr¸i theo thø tù sau
+ Th¨m gèc)

Nhóm B

1/ ý tưởng phương pháp sắp xếp chọn tăng dần (select sort)
" Chọn phần tử bé nhất xếp vào vị trí thứ nhất bằng cách đổi chổ phần tử bé nhất với phần tử thứ nhất; Tương tự đối với phần tử nhỏ thứ hai,ba..."

2/ ý tưởng phương pháp sắp xếp nổi bọt (bubble sort) là:
" Bắt đầu từ cuối dóy đến đầu dóy, ta lần lượt so sánh hai phần tử kế tiếp nhau, nếu phần tử nào nhỏ hơn được đứng vị trí trên."

3/ ý tưởng phương pháp sắp xếp chèn (insertion sort) là:
" Lần lượt lấy phần tử của danh sách chèn vị trí thích hợp của nó trong dóy bằng cóch đẩy các phần tử lớn hơn xuống."

4/ ý tưởng phương pháp sắp xếp nhanh (Quick sort) là:
" Lần lượt chia dãy phần tử thành hai dãy con bởi một phần tử khóa (dãy con trước khoá gồm các phần tử nhỏ hơn khoá và dãy còn lại gồm các phần tử lớn hơn khoá)."

5/ Phương pháp sắp xếp nhanh (Quick sort) chính là phương pháp:
" Phân đoạn"

6/ ý tưởng phương pháp sắp xếp Trộn (Merge sort) là:
" Phân đoạn dãy thành nhiều dãy con và lần lượt trộn hai dãy con thành dãy lớn hơn, cho đến khi thu được dãy ban đầu đã được sắp xếp."

7/ ý tưởng phương pháp sắp xếp vun đống (Heap sort) là:
" Lần lượt tạo đống cho cây nhị phân (phần tử gốc có giá trị lớn nhất) và loại phần tử gốc ra khỏi cây đưa vào dãy sắp xếp."

8/ Cơ chế heap trong sắp xếp vun đống là:

" Cây nhị phân đầy đủ với tính chất giá trị của nút cha luôn lớn hơn giá trị hai nút con."

9/ Trong giải thuật sắp xếp vun đống, ta có 4 thủ tục con (Insert - thêm 1 phần tử vào cây; Downheap - vun đống lại sau khi loại một phần tử khỏi Heap, Upheap- vun đống sau khi thêm một phần tử vào cây; Remove - loại 1 phần tử khỏi cây nhị phân). Để sắp xếp các phần tử trong dãy theo phương pháp vun đống, ta thực hiện 4 thủ tục trên theo thứ tự như thế nào?
" Insert – Upheap – Remove – Downheap"

10/ Tư tưởng của giải thuật tìm kiếm nhị phân:
" Tại mỗi bước tiến hành so sánh X với phần tử ở giữa của dãy. Dựa vào bước so sánh này quyết định giới hạn dãy tìm kiếm nằm ở nửa trên, hay nửa dưới của dãy hiện hành."

11/ Tư tưởng của giải thuật tìm kiếm tuần tự:
" So sánh X lần lượt với các phần tử thứ nhất, thứ hai,... của dãy cho đến khi gặp phần tử có khoá cần tõm."

12/ Tư tưởng của giải thuật tìm kiếm trên cây nhị phân tìm kiếm:
" Nếu giá trị cần tìm nhỏ hơn gốc thì thực hiện tìm kiếm trên cây con trái, ngược lại thì việc tìm kiếm được thực hiện trên cây con phải."

13/ Cây nhị phân tìm kiếm là:
Cây nhị phân mà mỗi nút trong cây đều thoả tính chất: giá trị của nút cha nhỏ hơn mọi nút trên cây con trái và lớn hơn mọi nút trên cây con phảI của nó."

14/ Trong các giải thuật sắp xếp, giải thuật nào áp dụng phương pháp Chia để trị?
" Quick sort, Merge sort"

15/ Thủ tục sau áp dụng giải thuật sắp xếp nào?

```
Procedure F
Begin
  For i:=1 to (n-1) do
   For j:=n downto (i+1) do _
       if a[j] < a[j-1] then
           begin   tg:=a[j];  a[j]:=a[j-1];  a[j-1]:=tg; end;
End;
```

Bubble sort

16/ Thủ tục sau áp dụng giải thuật sắp xếp nào?

```
Procedure F
Begin  a[0]:=- ∞;
        for i:=2 to n do
             begin    x:=a[i];   j:=i-1;
                      while x<a[j] do
                            begin    a[j+1]:=a[j]; j:=j-1;    end;
                       a[j+1]:=x;
            end;
  End;
```

Insert sort

17/ Thủ tục sau áp dụng giải thuật sắp xếp nào?

```
Procedure F(X,b,m,n,Z)
      Begin
            i:=k; i:=b; j:=m+1;
            while i<=m and j<=n do
                  if x[i] <=x[j] then
                        begin   z[k]:=x[i]; i:=i+1; end
                        else  begin  z[k]:=x[j]; j:=j+1; end;
                  k:=k+1;
                  if i>m then (zk,..,zn):= (xj,..,xn)
                        else (zk,..,zn):= (xi,..,xn)
      End;
```

Merge sort

18/ Thủ tục sau áp dụng giải thuật sắp xếp nào?

```
Procedure F(a, t, s);
Begin
      B:= true;
      if t<s then begin  i:=t; j:=s+1; key:=a[t];
                        while b do begin _
                            i:=i+1; while a[i]<=key do  i:=i+1;
                          j:=j -1; while a[j]>=key do j:=j-1;
                            if i<j then
                                begin    tg:=a[i]; a[i]:=a[j];    a[j]:=tg; end
                            else b:=false;
                            end;
                          tg:=a[t];  a[t]:=a[j];  a[j]:=tg;
                          call F(a, t,j-1);
                          cal  F(a, j+1,s);
                end;
      End;
```

Quick sort

19/ 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: {0 1 3 6 5 7 9 2 8 4}

20/ 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: {0 1 2 6 5 7 9 3 8 4}

21/ 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: {0 1 2 3 5 7 9 6 8 4}

22/ 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: {0 1 2 3 4 7 9 6 8 5}

23/ 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: {0 1 2 3 4 5 9 6 8 7}

24/ 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 bảy là: {0 1 2 3 4 5 6 9 8 7}

25/ Cho dãy số {6 1 3 0 5 7 9 2 8 4}. áp dụng phương pháp sắp xếp lựa chọn (Select sort) tăng dần, 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 tám là: {0 1 2 3 4 5 6 7 8 9}

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

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

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

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

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

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

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

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

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

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

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

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

38/ 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 hai là: {0 (1 2) 3 (5 4 8 6 9 7)}

39/ 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 ba là: {0 1 (2) 3 (5 4 8 6 9 7)}

40/ 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: {0 1 2 3 (4) 5 (8 6 9 7)}

41/ Cho dãy số: 12 2 8 5 1 6 4 15 và các bước sắp xếp sau:

Bước 1: 1 2 8 5 12 6 4 15
Bước 2: 1 2 8 5 12 6 4 15
Bước 3: 1 2 4 5 12 6 8 15
Bước 4: 1 2 4 5 12 6 8 15
Bước 5: 1 2 4 5 6 12 8 15
Bước 6: 1 2 4 5 6 8 12 15

Các bước trên dựa theo giải thuật sắp xếp nào?
Select sort

42/ Cho dãy số: 4 7 0 9 2 5 3 1 8 6 và các bước sắp xếp sau:

Bước 1: 0 4 7 1 9 2 5 3 6 8
Bước 2: 0 1 4 7 2 9 3 5 6 8
Bước 3: 0 1 2 4 7 3 9 5 6 8
Bước 4: 0 1 2 3 4 7 5 9 6 8
Bước 5: 0 1 2 3 4 5 6 7 8 9

Các bước trên dựa theo giải thuật sắp xếp nào?
Bubble sort

43/ Cho dãy số: 5 1 4 2 7 3 và các bước sắp xếp sau:

Bước 1: 1 5 4 2 7 3
Bước 2: 1 4 5 2 7 3
Bước 3: 1 2 4 5 7 3
Bước 4: 1 2 4 5 7 3
Bước 5: 1 2 3 4 5 7

Các bước trên dựa theo giải thuật sắp xếp nào?
"Insert sort"

44/ Cho dãy số 3 1 6 0 5 4 8 2 9 7 và các bước sắp xếp sau:

Bước 1: (0 1 2) 3 (5 4 8 6 9 7)
Bước 2: 0 (1 2) 3 (5 4 8 6 9 7)
Bước 3: 0 1 (2) 3 (5 4 8 6 9 7)
Bước 4: 0 1 2 3 (4) 5 (8 6 9 7)
Bước 5: 0 1 2 3 4 5 (8 6 9 7)
Bước 6: 0 1 2 3 4 5 (7 6) 8 (9)

Bước 7: 0 1 2 3 4 5 (6) 7 8 (9)
Bước 8: 0 1 2 3 4 5 6 7 8 (9)
Bước 9: 0 1 2 3 4 5 6 7 8 9
Các bước trên dựa theo giải thuật sắp xếp nào?
"Quick sort

45/ Cho dãy số : 3 1 6 0 5 4 8 2 9 7 và các bước sắp xếp sau:
Bước 1: 1 3 6 0 5 4 8 2 9 7
Bước 2: 1 3 6 0 5 4 8 2 9 7
Bước 3: 1 3 0 5 6 4 8 2 9 7
Bước 4: 0 1 3 5 6 4 8 2 9 7
Bước 5: 0 1 3 5 6 4 8 2 9 7
Bước 6: 0 1 3 5 6 4 8 2 7 9
Bước 7: 0 1 3 5 6 4 8 2 7 9
Bước 8: 0 1 3 5 6 2 4 7 8 9
Bước 9: 0 1 2 3 4 5 6 7 8 9
Các bước trên dựa theo giải thuật sắp xếp nào?
"Merge sort "

46/ Cho dãy số : 3 1 6 0 5 4 8 2 9 7 và các bước sắp xếp sau:
Bước 1: 1 3 0 6 4 5 2 8 7 9
Bước 2: 0 1 3 6 2 4 5 8 7 9
Bước 3: 0 1 2 3 4 5 6 8 7 9
Bước 4: 0 1 2 3 4 5 6 7 8 9
Các bước trên dựa theo giải thuật sắp xếp nào?
"Merge sort hai đường trực tiếp"

47/ Giải thuật sau thực hiện việc gì trong phương pháp sắp xếp vun đống?

```
Procedure F(v: integer)
Begin
        n:=n+1;
        a[n]:=v;
        upheap(n);
end;
```

Bổ sung một phần tử vào cây

48/ Giải thuật sau thực hiện việc gì trong phương pháp sắp xếp vun đống?

```
Procedure Upheap(k:integer);
Begin
      V:=a[k]; a[0]:=maxint;
      while a[k div 2] <= v do
            begin  a[k]:= a[k div 2]; k:=k div 2;  end;
      a[k]:=v;
End;
```

Vun đống cho cây sau khi thêm phần tử

49/ Giải thuật sau thực hiện việc gì trong phương pháp sắp xếp vun đống?

```
Procedure Downheap(k:integer)
Label 0;
```

```
Begin
    v:=a[k];
    While k<= n div 2 do
        begin j:=k*2;
            if a[j]<a[j+1] then j:=j+1;
            if v>=a[j] then goto 0;
            a[k]:=a[j]; k:=j;
        end;
    0: a[k]:=v;
End;
```

Vun đống cho cây sau khi loại bỏ phần tử

50/ Giải thuật sau thực hiện việc gì trong phương pháp sắp xếp vun đống?

```
Function P: integer;
Begin
    P:=a[1];
    a[1]:=a[n];
    n: =n-1;
    Downheap(1);
End;
```

Loại bỏ phần tử ra khỏi cây

Nhóm C:

1/ 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? 15 25 75 40 65 55 90 30 95 85

2/ 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? 15 25 75 40 65 55 90 30 95 85

3/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp lựa chọn, sau lượt 3 dãy sẽ được sắp xếp lại như thế nào? 15 25 30 40 65 55 90 75 95 85

4/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp lựa chọn, sau lượt 4 dãy sẽ được sắp xếp lại như thế nào? 15 25 30 40 65 55 90 75 95 85

5/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nổi bọt, sau lượt 1 dãy sẽ được sắp xếp lại như thế nào? 15 40 25 75 30 65 55 90 85 95

6/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nổi bọt, sau lượt 2 dãy sẽ được sắp xếp lại lại như thế nào? 15 25 40 30 75 55 65 85 90 95

7/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nổi bọt, sau lượt 3 dãy sẽ được sắp xếp lại như thế nào? 15 25 30 40 55 75 65 85 90 95

8/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nổi bọt, sau lượt 4 dãy sẽ được sắp xếp lại như thế nào? 15 25 30 40 55 75 65 85 90

9/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nhanh (Quick_Sort), sau lượt 1 dãy sẽ được sắp xếp lại như thế nào?

(15 25 30) 40 (65 55 90 75 95 85)

10/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nhanh (Quick_Sort), sau lượt 2 dãy sẽ được sắp xếp lại như thế nào?

1 5 (25 30) 40 (65 55 90 75 95 85)

11/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85.áp dụng phương pháp sắp xếp nhanh (Quick_Sort), sau lượt 3 dãy sẽ được sắp xếp lại như thế nào?

15 25 (30) 40 (65 55 90 75 95 85)

12/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nhanh (Quick_Sort), sau lượt 4 dãy sẽ được sắp xếp lại như thế nào?

15 25 30 40 (65 55 90 75 95 85)

13/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nhanh (Quick_Sort), sau lượt 5 dãy sẽ được sắp xếp lại như thế nào?

1 5 25 30 40 (55) 65 (90 75 95 85)

14/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nhanh (Quick_Sort), sau lượt 6 dãy sẽ được sắp xếp lại như thế nào?

1 5 25 30 40 55 65 (90 75 95 85)

15/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nhanh (Quick_Sort), sau lượt 7 dãy sẽ được sắp xếp lại như thế nào?

1 5 25 30 40 55 65 (85 75) 90 (95)

16/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nhanh (Quick_Sort), sau lượt 8 dãy sẽ được sắp xếp lại như thế nào?

1 5 25 30 40 55 65 (75) 85 90 (95)

17/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp nhanh (Quick_Sort), sau lượt 9 dãy sẽ được sắp xếp lại như thế nào?

1 5 25 30 40 55 65 7 5 85 90 (95)

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

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

19/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp hòa nhập (Merge_Sort), sau lượt 2 dãy sẽ được sắp xếp lại như thế nào?

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

20/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp hòa nhập (Merge_Sort), sau lượt 3 dãy sẽ được sắp xếp lại như thế nào?

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

21/ Cho dãy số sau: 40 25 75 15 65 55 90 30 95 85. áp dụng phương pháp sắp xếp hòa nhập (Merge_Sort), sau lượt 4 dãy sẽ được sắp xếp lại như thế nào?

15 25 30 40 55 65 75 85 90 95

22/ 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? 4

23/ Cho dãy số sau: 14 32 10 43 57 87 55 36 97 11. áp dụng phương pháp tìm kiếm nhị phân, sau bao nhiêu lần phân đoạn ta sẽ tìm thấy số 43? 3

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

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

26/ Tính chất nào sau đây là tính chất của cây nhị phân tìm kiếm:

Mọi khóa thuộc cây con trái nút đó đều nhỏ hơn khóa ứng với nút đó

27/ Tính chất nào sau đây là tính chất của cây nhị phân tìm kiếm?

Mọi khóa thuộc cây con phải nút đó đều lớn hơn khóa ứng với nút đó

28/ Cho cây nhị phân tìm kiếm sau:Nếu tìm số 50 thì ta phải thực hiện phép so sánh bao nhiêu lần: (tùy hình)

29/ Giải thuật sau là phương pháp tìm kiếm nào?

```
Function F(x)
Begin
      i:=1; a[n+1]:=x;
      while a[i] <> x do i:=i+1;
      if  i=n+1 then  return(0)
      else return(i);
End;
```

Tìm kiếm tuần tự

30/ Giải thuật sau là phương pháp tìm kiếm nào?

```
Function Binary_search(l,r,x)
Begin
       If l>r then  k:=0
      Else  m:= (l+r) div 2
              If x< a[m]  then K:=binary_search(l, m, x)
              Else  If x>a[m] then K:=binary_search(m+1,r,x)
                    Else k:=m;
       Return(m);
End;
```

Tìm kiếm nhị phân

31/ Định nghĩa đống: ***Heap*** *(Đống) là một cây nhị phân thoả mãn 2 tính chất sau:*

- *Giá trị khoá của nút cha luôn luôn lớn hơn khoá của tất cả các nút con.*

- *Heap phải là một cây nhị phân hoàn chỉnh.*

32/ Định nghĩa cây nhị phân tìm kiếm ứng với n khóa k1,..,kn:
k[2i] < k[i] <k[2i+1] với mọi i = 1→n

33/ Điều kiện của một cây nhị phân tìm kiếm: *là cây nhị phân hoàn chỉnh (đầy đủ) và mỗi nút trong cây đều thoả tính chất: giá trị của nút cha nhỏ hơn mọi nút trên cây con trái và lớn hơn mọi nút trên cây con phải của nó.*

34/ Cho giải thuật tìm kiếm phần tử trong cây nhị phân tìm kiếm có gốc k:

```
Procedure tree_search(k,x);
Begin
If (a[k]=null) or (a[k]=x) then return(k)
Else
If  x<a[k] then tree_search(2*k,x)
Else  tree_search(2*k+1,x);
End;
```

Cho biết câu lệnh điều dừng đệ quy? *If (a[k]=null) or (a[k]=x) then return(k)*

35/ Cho giải thuật tìm kiếm phần tử trong cây nhị phân tìm kiếm có gốc k:

```
Procedure tree_search(k,x);
Begin
If (a[k]=null) or (a[k]=x) then return(k)
Else
If  x<a[k] then tree_search(2*k,x)
Else  tree_search(2*k+1,x);
End;
```

Khi phần tử x cần tìm bằng a[k] thì : *thủ tục trả về giá trị k → tìm thấy*

36/ Cho giải thuật tìm kiếm phần tử trong cây nhị phân tìm kiếm có gốc k:

```
Procedure tree_search(k,x);
Begin
If (a[k]=null) or (a[k]=x) then return(k)
Else
If  x<a[k] then tree_search(2*k,x)
Else  tree_search(2*k+1,x);
End;
```

Khi phần tử x< a[k] thì: *thực hiện lệnh tìm trên cây con trái : tree_search(2*k,x)*

37/ Cho giải thuật tìm kiếm phần tử trong cây nhị phân tìm kiếm có gốc k:

```
Procedure tree_search(k,x);
Begin
If (a[k]=null) or (a[k]=x) then return(k)
Else
If  x<a[k] then tree_search(2*k,x)
Else  tree_search(2*k+1,x);
End;
```

Lời gọi tìm kiếm x trong cây trái: tree_search(2*k,x)

38/ Cho giải thuật tìm kiếm phần tử trong cây nhị phân tìm kiếm có gốc k:

```
Procedure tree_search(k,x);
Begin
If (a[k]=null) or (a[k]=x) then return(k)
Else
If  x<a[k] then tree_search(2*k,x)
Else  tree_search(2*k+1,x);
End;
```

Khi phần tử x> a[k] thì: *thực hiện lệnh tìm trên cây con phải: tree_search(2*k+1,x);*

39/ Cho giải thuật tìm kiếm phần tử trong cây nhị phân tìm kiếm có gốc k:

```
Procedure tree_search(k,x);
Begin
If (a[k]=null) or (a[k]=x) then return(k)
Else
If  x<a[k] then tree_search(2*k,x)
Else  tree_search(2*k+1,x);
End;
```

Lời gọi tìm kiếm x trong cây phải: tree_search(2*k+1,x);

40/ Cho giải thuật tìm kiếm phần tử trong cây nhị phân tìm kiếm có gốc k:

```
Procedure tree_search(k,x);
Begin
If (a[k]=null) or (a[k]=x) then return(k)
Else
If  x<a[k] then tree_search(2*k,x)
Else  tree_search(2*k+1,x);
End;
```

Độ phức tạp thuật toán trên là: "O(logn)."

41/ Cho giải thuật sắp xếp lựa chọn:

```
Procedure Select_sort(a[1], a[2],…,a[n], n);
Begin
for i:=1 to n-1 do
  for j:=i+1 to n do
    if a[i]>a[j] then
      begin
            tg:=a[i]; a[i]:=a[j]; a[j]:=tg;
       end;
End;
```

Vòng for i:=1 to n-1 do có tác dụng: *Xác định vị trí i cần đổi chỗ (xác định số lượt sắp xếp)*

42/ Cho giải thuật sắp xếp lựa chọn:

```
Procedure Select_sort(a[1], a[2],…,a[n], n);
Begin
for i:=1 to n-1 do
  for j:=i+1 to n do
    if a[i]>a[j] then
```

```
            begin
                tg:=a[i]; a[i]:=a[j]; a[j]:=tg;
             end;
         End;
```
với n >3, khi i =3 thì dãy phần tử được lấy ra để tìm giá trị nhỏ nhất đưa lên vị trí đầu là:
bắt đầu từ vị trí 4 đến vị trí n

43/ Cho giải thuật sắp xếp lựa chọn:
```
Procedure Select_sort(a[1], a[2],…,a[n], n);
Begin
for i:=1 to n-1 do
 for j:=i+1 to n do
  if a[i]>a[j] then
   begin
      tg:=a[i]; a[i]:=a[j]; a[j]:=tg;
    end;
End;
```
với n >3, khi i =3 thì dãy ban đầu các phần tử nào đó được sắp xếp? *phần tử tại vị trí 1 và 2*

44/ Cho giải thuật sắp xếp lựa chọn:
```
Procedure Select_sort(a[1], a[2],…,a[n], n);
Begin
for i:=1 to n-1 do
 for j:=i+1 to n do
  if a[i]>a[j] then
   begin
      tg:=a[i]; a[i]:=a[j]; a[j]:=tg;
    end;
End;
```
Tác dụng của vòng lặp: for j:=i+1 to n do : *tìm phần tử thứ j nhỏ nhất để đổi chỗ*

45/ Cho giải thuật sắp xếp lựa chọn:
```
Procedure Select_sort(a[1], a[2],…,a[n], n);
Begin
for i:=1 to n-1 do
 for j:=i+1 to n do
  if a[i]>a[j] then
   begin
      tg:=a[i]; a[i]:=a[j]; a[j]:=tg;
    end;
End;
```
các lệnh : tg:=a[i]; a[i]:=a[j]; a[j]:=tg; có tác dụng: *đổi chỗ vị trí của a[i] với a[j]*

46/ Cho giải thuật sắp xếp nổi bọt:
```
Procedure BUBBLE_SORT(a[1], a[2],…,a[n], n);
Begin
For i:=1 to (n-1) do                (1)
    For j:=n downto (i+1) do        (2)
        If a[j] < a[j-1] then       (3)
```

```
                    begin
            tg:=a[j]; a[j]:=a[j-1]; a[j-1]:=tg; (4)
             end ;
        End ;
```
Lệnh (1) có tác dụng: *Xác định số lượt sắp xếp.*

47/ Cho giải thuật sắp xếp nổi bọt:
```
        Procedure BUBBLE_SORT(a[1], a[2],…,a[n], n);
        Begin
        For i:=1 to (n-1) do   (1)
            For j:=n downto (i+1) do (2)
                If a[j] < a[j-1] then (3)
                    begin
            tg:=a[j]; a[j]:=a[j-1]; a[j-1]:=tg; (4)
             end ;
        End ;
```
Vòng (1) và (2) hoạt động: *duyệt từ đầu dãy đến cuối dãy, tại lượt thứ i lại duyệt từ vị trí cuối dãy lên đến vị trí i+1*

48/ Cho giải thuật sắp xếp nổi bọt:
```
        Procedure BUBBLE_SORT(a[1], a[2],…,a[n], n);
        Begin
        For i:=1 to (n-1) do          (1)
            For j:=n downto (i+1) do (2)
                If a[j] < a[j-1] then (3)
                    begin
            tg:=a[j]; a[j]:=a[j-1]; a[j-1]:=tg; (4)
             end ;
        End ;
```
Vòng (1) và (2) hoạt động:

49/ Cho giải thuật sắp xếp nổi bọt:
```
        Procedure BUBBLE_SORT(a[1], a[2],…,a[n], n);
        Begin
        For i:=1 to (n-1) do  ) (1)
            For j:=n downto (i+1) do (2)
                If a[j] < a[j-1] then (3)
                    begin
            tg:=a[j]; a[j]:=a[j-1]; a[j-1]:=tg; (4)
             end ;
```
Vòng (2) và (3) và (4) hoạt động: *Kiểm tra phần tử nào nhỏ hơn phía cuối thì đổi chỗ đẩy lên phía đầu dãy.*

50/ Cho giải thuật sắp xếp Quick sort:
```
        Procedure Quick_sort (t, s);
          var tg, i j : integer;
          Begin
            B:= true;
            if t<s then  (1)
```

```
            begin
               i:=t;   j:=s+1; key:=a[t];
               while B do (2)
                  begin
                     i:=i+1;
                     while a[i]<=key do  i:=i+1; (3)
                     j:=j-1;
                     while a[j]>=key do j:=j-1; (4)
                     if i<j then
                     begin   tg:=a[i]; a[i]:=a[j]; a[j]:=tg; end (5)
                     else B:=false;
                  end;
               tg:=a[t]; a[t]:=a[j]; a[j]:=tg; (6)
               call QUICK_SORT(t,j-1); (7)
               cal QUICK_SORT(j+1,s); (8)
            end;
      End;
```

Điều kiện (1) đúng nghĩa là: *Kiểm tra từ vị trí sau vị trí khóa*

51/ Cho giải thuật sắp xếp Quick sort:

```
      Procedure Quick_sort (t, s);
        var tg, i j : integer;
        Begin
          B:= true;
          if t<s then  (1)
            begin
               i:=t;   j:=s+1; key:=a[t];
               while B do (2)
                  begin
                     i:=i+1;
                     while a[i]<=key do  i:=i+1; (3)
                     j:=j-1;
                     while a[j]>=key do j:=j-1; (4)
                     if i<j then
                     begin   tg:=a[i]; a[i]:=a[j]; a[j]:=tg; end (5)
                     else B:=false;
                  end;
               tg:=a[t]; a[t]:=a[j]; a[j]:=tg; (6)
               call QUICK_SORT(t,j-1); (7)
               cal QUICK_SORT(j+1,s); (8)
            end;
      End;
```

Biến a[i] là đại diện cho : *các phần tử đầu dãy nằm trước khóa (nhỏ hơn khóa)*

52/ Cho giải thuật sắp xếp Quick sort:

```
      Procedure Quick_sort (t, s);
        var tg, i j : integer;
        Begin
          B:= true;
```

```
            if t<s then  (1)
               begin
                  i:=t;   j:=s+1; key:=a[t];
                  while B do (2)
                     begin
                        i:=i+1;
                        while a[i]<=key do  i:=i+1; (3)
                        j:=j-1;
                        while a[j]>=key do j:=j-1; (4)
                        if i<j then
                        begin   tg:=a[i]; a[i]:=a[j]; a[j]:=tg; end (5)
                        else B:=false;
                     end;
                  tg:=a[t]; a[t]:=a[j]; a[j]:=tg; (6)
                  call QUICK_SORT(t,j-1); (7)
                  cal QUICK_SORT(j+1,s); (8)
               end;
         End;
```

Biến a[j] là đại diện cho : *Các phần tử nằm cuối dãy (lớn hơn khóa)*

53/ Cho giải thuật sắp xếp Quick sort:

```
      Procedure Quick_sort (t, s);
         var tg, i j : integer;
         Begin
            B:= true;
            if t<s then  (1)
               begin
                  i:=t;   j:=s+1; key:=a[t];
                  while B do (2)
                     begin
                        i:=i+1;
                        while a[i]<=key do  i:=i+1; (3)
                        j:=j-1;
                        while a[j]>=key do j:=j-1; (4)
                        if i<j then
                        begin   tg:=a[i]; a[i]:=a[j]; a[j]:=tg; end (5)
                        else B:=false;
                     end;
                  tg:=a[t]; a[t]:=a[j]; a[j]:=tg; (6)
                  call QUICK_SORT(t,j-1); (7)
                  cal QUICK_SORT(j+1,s); (8)
               end;
         End;
```

Vòng lặp (2) vẫn thực hiện khi : i<j

54/ Cho giải thuật sắp xếp Quick sort:

```
      Procedure Quick_sort (t, s);
         var tg, i j : integer;
         Begin
```

```
            B:= true;
            if t<s then  (1)
               begin
                  i:=t;   j:=s+1; key:=a[t];
                  while B do (2)
                     begin
                        i:=i+1;
                        while a[i]<=key do  i:=i+1; (3)
                        j:=j-1;
                        while a[j]>=key do j:=j-1; (4)
                        if i<j then
                        begin   tg:=a[i]; a[i]:=a[j]; a[j]:=tg; end (5)
                        else B:=false;
                     end;
                  tg:=a[t]; a[t]:=a[j]; a[j]:=tg; (6)
                  call QUICK_SORT(t,j-1); (7)
                  cal QUICK_SORT(j+1,s); (8)
               end;
         End;
```

(6) có tác dụng : *đổi chỗ phần tử a[j] cho khóa. (đổi chỗ vị trí khóa vào vị trí j)*

55/ Cho giải thuật sắp xếp Quick sort:

```
      Procedure Quick_sort (t, s);
         var tg, i j : integer;
         Begin
            B:= true;
            if t<s then  (1)
               begin
                  i:=t;   j:=s+1; key:=a[t];
                  while B do (2)
                     begin
                        i:=i+1;
                        while a[i]<=key do  i:=i+1; (3)
                        j:=j-1;
                        while a[j]>=key do j:=j-1; (4)
                        if i<j then
                        begin   tg:=a[i]; a[i]:=a[j]; a[j]:=tg; end (5)
                        else B:=false;
                     end;
                  tg:=a[t]; a[t]:=a[j]; a[j]:=tg; (6)
                  call QUICK_SORT(t,j-1); (7)
                  cal QUICK_SORT(j+1,s); (8)
               end;
         End;
```

(7) có tác dụng : thực hiện sắp xếp dãy nhỏ hơn khóa

56/ Cho giải thuật sắp xếp Quick sort:

```
Procedure Quick_sort (t, s);
  var tg, i j : integer;
```

```
Begin
   B:= true;
   if t<s then  (1)
      begin
         i:=t;   j:=s+1; key:=a[t];
         while B do (2)
            begin
               i:=i+1;
               while a[i]<=key do  i:=i+1; (3)
               j:=j-1;
               while a[j]>=key do j:=j-1; (4)
               if i<j then
               begin   tg:=a[i]; a[i]:=a[j]; a[j]:=tg; end (5)
               else B:=false;
            end;
         tg:=a[t]; a[t]:=a[j]; a[j]:=tg; (6)
         call QUICK_SORT(t,j-1); (7)
         cal QUICK_SORT(j+1,s); (8)
      end;
End;
```

(8) có tác dụng : *thực hiện sắp xếp dãy lớn hơn khóa*

57. Khi lưu trữ cây nhị phân dưới dạng mảng, nếu vị trí của nút cha trong mảng là 6 thì vị trí tương ứng của nút con sẽ là: 12, 13

58. Khi lưu trữ cây nhị phân dưới dạng mảng, nếu vị trí của nút con trong mảng là 9 thì vị trí tương ứng của nút cha sẽ là: 4

59. Khi lưu trữ cây nhị phân dưới dạng mảng, nếu vị trí của nút cha trong mảng là i thì vị trí của nút con trái là: 2*i

60. Khi lưu trữ cây nhị phân dưới dạng mảng, nếu vị trí của nút cha trong mảng là i thì vị trí của nút con phải là: 2*i+1

61. Cho cây nhị phân: A, B, C, D, E, F, G, H, I, J, K, L, M, N. Cây con trái của cây A:
B, D, E, H, I, J, K

62. Cho cây nhị phân: A, B, C, D, E, F, G, H, I, J, K, L, M, N. Các nút lá:
H, I, J, K, L, M, N

63. Cho cây nhị phân: A, B, C, D, E, F, G, H, I, J, K, L, M, N. Cây con trái của cây B:
D, H, I

64. Cho cây nhị phân: A, B, C, D, E, F, G, H, I, J, K, L, M, N. Cây con trái của cây C:
F, L, M

65. Cho cây nhị phân: A, B, C, D, E, F, G, H, I, J, K, L, M, N. Cây con phải của cây C:
G, N

66. Cho cây nhị phân: A, B, C, D, E, F, G, H, I, J, K, L, M, N. Cây con phải của cây B: E, J, K

67. Cho cây nhị phân: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16. Phần tử con của 4: 9, 10

68. Cho cây nhị phân: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16. Phần tử cha của 4: 3

69. Cho cây nhị phân: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16. Phần tử con của 2: 24, 5

70. Cho cây nhị phân: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16. Phần tử cha của 2: 6

71. Cho cây nhị phân là: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Áp dụng giải thuật duyệt cây nhị phân theo thứ tự giữa cho cây có gốc là 3, thỡ thứ tự các phần tử được thăm là: 1, 7, 21, 3, 10, 4, 9

72. Cho cây nhị phân là: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Áp dụng giải thuật duyệt cây nhị phân theo thứ tự trước cho cây có gốc là 6, thì thứ tự các phần tử được thăm là: 6, 2, 24, 5, 8, 17, 16

73. Cho cây nhị phân là: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Áp dụng giải thuật duyệt cây nhị phân theo thứ tự giữa cho cây có gốc là 6, thì thứ tự các phần tử được thăm là: 24, 2, 5, 6, 17, 8, 16

74. Cho cây nhị phân là: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Áp dụng giải thuật duyệt cây nhị phân theo thứ tự sau cho cây có gốc là 6, thỡ thứ tự các phần tử được thăm là: 24, 5, 2, 17, 16, 8, 6

75. Giải thuật:

```
procedure preorder (k);
    begin
        if a[k] <> null then
        begin
            write(a[k]);        (1)
            preorder(2*k);      (2)
            preorder(2*k+1);    (3)
        end;
     end;
```

(2) là: *Duyệt cây con trái theo thứ tự trước*

76. Giải thuật:

```
procedure preorder (k);
    begin
        if a[k] <> null then
        begin
            write(a[k]);        (1)
```

```
                preorder(2*k);          (2)
                preorder(2*k+1);        (3)
            end;
        end;
```
(3) là: *Duyệt cây con phải theo thứ tự trước*

77. Giải thuật:
```
    procedure preorder (k);
        begin
            if a[k] <> null then
            begin
                write(a[k]);            (1)
                preorder(2*k);          (2)
                preorder(2*k+1);        (3)
            end;
        end;
```
Cho cây: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16. Nếu áp dụng giải thuật trên với k=1, thì lời gọi (3) duyệt cây có gốc là: 6

78. Giải thuật:
```
    procedure preorder (k);
        begin
            if a[k] <> null then
            begin
                write(a[k]);            (1)
                preorder(2*k);          (2)
                preorder(2*k+1);        (3)
            end;
        end;
```
Cho cây: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16. Nếu áp dụng giải thuật trên với k=1, thì lời gọi (2) duyệt cây có gốc là: 3

79. Giải thuật sau là:
```
    procedure Preorder (k);
        begin
            if a[k] <> null then
            begin
                write(a[k]);
                Preorder(2*k);
                Preorder(2*k+1);
            end;
        end;
```
Duyệt cây nhị phân theo thứ tự trước

80. Giải thuật:
```
    procedure Preorder (k);
        begin
            if a[k] <> null then
            begin
```

```
                write(a[k]);
                Preorder(2*k);
                Preorder(2*k+1);
        end;
    end;
```

Nếu phần tử trong cây nhị phân là:
25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Nếu áp dụng giải thuật trên thì thứ tự các phần tử sẽ là: 25, 3, 7, 1, 21, 4, 10, 9, 6, 2, 24, 5, 8, 17, 16.

81. Giải thuật:

```
procedure Preorder (k);
    begin
        if a[k] <> null then
        begin
            write(a[k]);
            Preorder(2*k);
            Preorder(2*k+1);
        end;
    end;
```

Nếu phần tử trong cây nhị phân là:
25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Khi áp dụng giải thuật trên, với k = 2 thì cây được duyệt là: 3, 7, 1, 21, 4, 10, 9

82. Giải thuật:

```
procedure Preorder (k);
    begin
        if a[k] <> null then
        begin
            write(a[k]);
            Preorder(2*k);
            Preorder(2*k+1);
        end;
    end;
```

Nếu phần tử trong cây nhị phân là:
25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Khi áp dụng giải thuật trên, với k = 3 thì cây được duyệt là: 6, 2, 24, 5, 8, 17, 16.

83. Giải thuật:

```
procedure Preorder (k);
    begin
        if a[k] <> null then
        begin
            write(a[k]);
            Preorder(2*k);
            Preorder(2*k+1);
        end;
    end;
```

Nếu phần tử trong cây nhị phân là:

25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Khi áp dụng giải thuật trên, với k = 4 thì cây được duyệt là: 7, 1, 21

84. Giải thuật:

```
procedure Preorder (k);
    begin
        if a[k] <> null then
        begin
            write(a[k]);
            Preorder(2*k);
            Preorder(2*k+1);
        end;
     end;
```

Nếu phần tử trong cây nhị phân là:
25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Khi áp dụng giải thuật trên, với k = 5 thì cây được duyệt là: 4, 10, 9

85. Giải thuật:

```
procedure Preorder (k);
    begin
        if a[k] <> null then
        begin
            write(a[k]);
            Preorder(2*k);
            Preorder(2*k+1);
        end;
     end;
```

Nếu phần tử trong cây nhị phân là:
25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Khi áp dụng giải thuật trên, với k = 6 thì cây được duyệt là: 2, 24, 5

86. Giải thuật:

```
procedure Preorder (k);
    begin
        if a[k] <> null then
        begin
            write(a[k]);
            Preorder(2*k);
            Preorder(2*k+1);
        end;
     end;
```

Nếu phần tử trong cây nhị phân là:
25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Khi áp dụng giải thuật trên, với k = 7 thì cây được duyệt là: 8, 17, 16

87. Duyệt cây nhị phân theo thứ tự trước được thực hiện theo thứ tự: *Duyệt gốc, duyệt cây con trái theo thứ tự trước, duyệt cây con phải theo thứ tự trước.*

88. Duyệt cây nhị phân theo thứ tự giữa được thực hiện theo thứ tự: *Duyệt cây con trái theo thứ tự giữa, duyệt gốc, duyệt cây con phải theo thứ tự giữa.*

89. Duyệt cây nhị phân theo thứ tự sau được thực hiện theo thứ tự: *Duyệt cây con trái theo thứ tự sau, duyệt cây con phải theo thứ tự giữa, duyệt gốc.*

90. Có bao nhiêu cách duyệt cây nhị phân ? 3

91. Cho cây nhị phân: A B C D E F. Sử dụng phép duyệt theo thứ tự trước với cây trên cho kết quả thứ tự các phần tử: A, B, D, E, C, F

92. Cho cây nhị phân: A B C D E F. Sử dụng phép duyệt theo thứ tự giữa với cây trên cho kết quả thứ tự các phần tử: D, B, E, A, C, F

93. Cho cây nhị phân: A B C D E F. Sử dụng phép duyệt theo thứ tự sau với cây trên cho kết quả thứ tự các phần tử: D, E, B, F, C, A

94. Giải thuật:

```
procedure Inorder (k);
    begin
        if a[k] <> null then
        begin
            Inorder(2*k);
            write(a[k]);
            Inorder(2*k+1);
        end;
     end;
```

Là phép duyệt cây theo: *thứ tự giữa*

95. Giải thuật:

```
procedure Postorder (k);
    begin
        if a[k] <> null then
        begin
            Write(a[k]);
            Postorder(2*k);
            Postorder(2*k+1);
        end;
     end;
```

Là phép duyệt cây theo: *thứ tự trước*

96. Giải thuật:

```
procedure Postorder (k);
    begin
        if a[k] <> null then
        begin
            Write(a[k]);
            Postorder(2*k);
            Postorder(2*k+1);
```

```
            end;
        end;
```
Nếu phần tử trong cây nhị phân là:
25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Khi áp dụng giải thuật trên, thứ tự các phần tử được thăm là: 25, 3, 7, 1, 21, 4, 10, 9, 6, 2, 24, 5, 8, 17, 16

97. Giải thuật:
```
    procedure Inorder (k);
        begin
            if a[k] <> null then
            begin
                Inorder(2*k);
                write(a[k]);
                Inorder(2*k+1);
            end;
        end;
```
Nếu phần tử trong cây nhị phân là:
25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Khi áp dụng giải thuật trên, thứ tự các phần tử được thăm là:
1, 7, 21, 3, 10, 4, 9, 25, 24, 2, 5, 6, 17, 8, 16

98. Cho cây nhị phân là: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Áp dụng giải thuật duyệt cây nhị phân theo thứ tự trước cho cây có gốc là 3, thỡ thứ tự các phần tử được thăm là: 3, 7, 1, 21, 4, 10, 9

99. Cho cây nhị phân là: 25 3 6 7 4 2 8 1 21 10 9 24 5 17 16
Áp dụng giải thuật duyệt cây nhị phân theo thứ tự sau cho cây có gốc là 3, thỡ thứ tự các phần tử được thăm là: 1, 21, 7, 10, 9, 4, 3

100. 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? 4

101. 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? 3

102. 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 số là: 10, 11, 14, 32

103. 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 97, lần phân đoạn thứ hai của dãy số là: 87, 97

104. Tính chất nào sau đây là tính chất của cây nhị phân tìm kiếm:
"Cây nhị phân mà mỗi nút trong cây đều thoả tính chất: giá trị của nút cha nhỏ hơn mọi nút trên cây con trái và lớn hơn mọi nút trên cây con phải của nó."

105. Tính chất nào sau đây là tính chất của cây nhị phân tìm kiếm?

106. Giải thuật sau là phương pháp tìm kiếm nào?
```
    Function F(x)
```

```
Begin
  i:=1; a[n+1]:=x;
  while a[i] <> x do i:=i+1;
  if  i=n+1 then  return(0)
  else return(i);
End;
```

Tìm kiếm tuần tự

107. Giải thuật sau là phương pháp tìm kiếm nào?

```
Function Binary_search(l,r,x)
Begin
  If l>r then  k:=0
       Else  m:= (l+r) div 2
     If x< a[m]  then K:=binary_search(l, m, x)
     Else  If x>a[m] then K:=binary_search(m+1,r,x)
           Else k:=m;
  Return(m);
End;
```

Tìm kiếm nhị phân

108 .Cho giải thuật tìm kiếm:

```
Function Binary_search(l,r,x)
Begin
  If l>r then  k:=0
       Else  m:= (l+r) div 2
     If x< a[m]  then K:=binary_search(l, m, x)
     Else  If x>a[m] then K:=binary_search(m+1,r,x)
           Else k:=m;
  Return(m);
End;
```

Sau khi chia dãy số thành 2 dãy con, phần tử đầu của dãy thứ 2 là:
a[m+1]

109. 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 trái sẽ là: 6

110. 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 là: 7