

I. Bài Tập Lý Thuyết

Phần I. Độ phức tạp thuật toán

1) Tính độ phức tạp của các khối lệnh sau:

- a)

```
tong = 0;
for(i=0; i<n; i++)
    for(j=i; j<=i+3; j++)
        tong += j;
```
- b)

```
tich = 0;
for(i=n; i>0; i--)
    for(j= 0; j<2*i; j++)
        tich *= j;
```
- c)

```
tich= 0;
for(i=n; i>=1; i--)
    for(j=i; j>=1; j--)
        for(k=i; k<=i+3; k++)
            tong+= i+ j+ k;
```
- d)

```
tong= 0;
for(i=1; i<=n; i++)
    for(j=1; j<=i; j++)
        for(k=1; k<=2*i; k++)
            tong+= i+ j+ k;
```
- e)

```
tong= 0;
for(i=1; i<=n; i++)
    for(j=1; j<=i; j++)
        for(k=1; k<=j; k++)
            tong+= i+ j+ k;
```
- f)

```
for(i=0; i<n-1; i++)
    for(j=i+1; j<n; j++)
        if (a[j]< a[i])
        {
            int t = a[i];
            a[i]= a[j];
            a[j]= t;
        }
```

2) Tính độ phức tạp của chương trình đổi số thập phân sang nhị phân sau:

```
void NhiPhan(int n)
{
    int c= 0, a[100];
    while (n>0)
    {
        c++;
        a[c]= n%2;
        n= n/2;
    }
    printf("Ket qua la: ");
    for(int i=c; i>=1; i--)
        printf("%d", a[i]);
}
```

Phần II. Giải thuật đệ quy

- 1) Viết chương trình đệ quy tính giai thừa của số nguyên n ($n!$) nhập từ bàn phím.
- 2) Viết chương trình đệ quy tìm số Fibonacci thứ n với n nhập từ bàn phím.

$$F_1 = F_2 = 1$$

$$F_n = F_{n-1} + F_{n-2}$$

- 3) Cho tổ hợp: $C_n^k = \frac{n!}{k!(n-k)!}$ với $(0 < k < n)$

Người ta đã chứng minh tổ hợp trên cũng có thể tính bằng đệ quy qua công thức sau:

- $C_n^n = C_n^0 = 1$

- $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$

Hãy viết chương trình tính C_n^k theo cách tính đệ quy.

- 4) Viết chương trình phân tích số n thành tích các thừa số nguyên tố.
Vd: $90 = 2 * 3 * 3 * 5$.
- 5) Viết chương trình tìm USCLN của 2 số bằng đệ quy.
- 6) Viết chương trình tính lũy thừa x^n (x số thực, n số nguyên) bằng phương pháp đệ quy.

Phần III. Mảng và sắp xếp mảng

- 1) Cho mảng a sau bắt đầu tại địa chỉ 30.
`int a[4][5];`
hãy tính địa chỉ của phần tử $a[1][4]$, phần tử $a[3][2]$
- 2) Viết hàm đảo ngược mảng số nguyên.
Ví dụ: 1 5 4 7 \rightarrow 7 4 5 1
- 3) Viết hàm dịch trái mảng số nguyên.
Ví dụ: 1 5 4 7 \rightarrow 5 4 7 1
- 4) Viết xóa một phần tử tại vị trí thứ i trong mảng.
Ví dụ: 1 5 4 7 \rightarrow 5 4 7 1
- 5) Minh họa thuật toán SelectionSort cho dãy
8 3 7 1 9 4 6
- 6) Minh họa thuật toán InsertionSort cho dãy
31 08 12 19 17 09 02 33 25
- 7) Minh họa thuật toán BubbleSort cho dãy
31 08 12 19 17 09
- 8) Minh họa phương pháp phân hoạch Lomuto
49 09 35 07 82 19 73 59 47 15 38

Phần IV. Con trỏ và danh sách liên kết.

1) So sánh 3 hàm hoán vị sau.

```
void Swap1(int x, int y)
{
    int temp= x;
    x= y;
    y= temp;
}
void Swap2(int &x, int &y)
{
    int temp = x;
    x=y;
    y = temp;
}
void Swap3(int *x, int *y)
{
    int temp = *x;
    *x=*y;
    *y = temp;
}
void main()
{
    int a, b;
    a=5; b=7;
    Swap1(a, b);
    a=5; b=7;
    Swap2(a, b);
    a=5; b=7;
    Swap3(&a, &b);
}
```

Hàm nào thật sự hoán vị a và b?

2) Xét chương trình.

```
void cong(int *x, int *y)
{
    *x = *x + 1;
    *y = *y + 1;
}
void main()
{
    int a = 5;
    cong(&a, &a);
    printf("a= %d", a);
}
```

Cho biết kết quả in ra màn hình của a. Giải thích.

3) Xét chương trình.

```
void SoSanh(int *p, int *q)
{
    *p= *p+ 1;
    int x= *p, y= *q;
    if (x!=y) printf("x, y khác nhau");
    else printf("x, y bằng nhau");
}
```

```

void main()
{
    int a= 5;
    SoSanh(&a, &a);
}

```

Cho biết kết quả in ra màn hình. Giải thích.

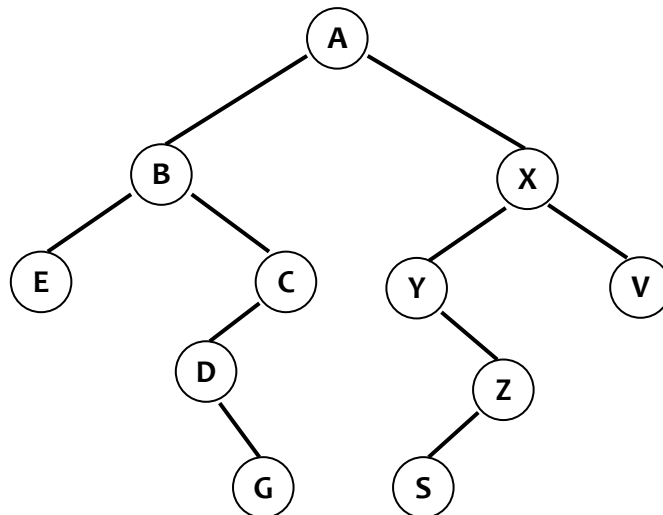
- 4) Viết hàm đếm số phần tử có trong danh sách liên kết đơn.
- 5) Viết hàm tính tổng các phần tử có trong danh sách liên kết đơn gồm các số nguyên
- 6) Viết hàm kiểm tra xem danh sách liên kết gồm các số nguyên có phải là danh sách sắp thứ tự từ bé đến lớn.

Phần V. Stack và queue.

- 1) Dùng stack tính giá trị của các biểu thức sau:
 - a. $2\ 4\ 3\ -\ 5\ *\ +$
 - b. $1\ 2\ 3\ +\ 4\ *\ -\ 5\ /\ 6\ *$
- 2) Chuyển đổi các biểu thức sau từ dạng trung tố thành hậu tố:
 - a) $A / (B + C * D - E) * F$
 - b) $(A - B) * (C - D ^ E * F) + G$

Phần VI. Cây nhị phân.

- 1) Cho biết kết quả duyệt cây sau theo thứ tự NLR, LNR và LRN



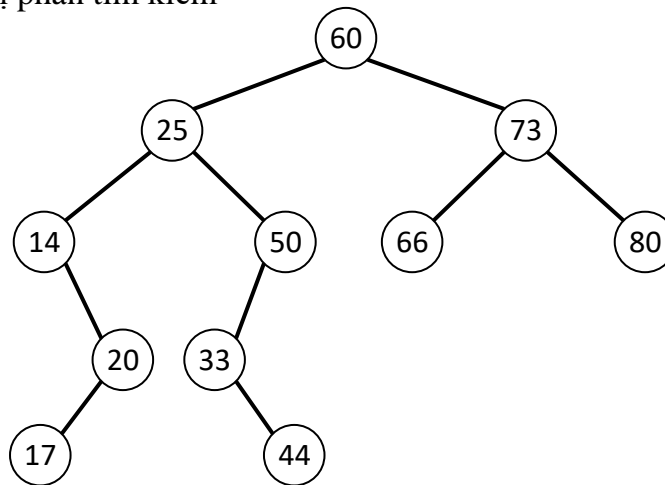
- 2) Vẽ cây biểu diễn biểu thức : $((a+b)+c(d-e)+f)*(g-h)^2$
Cho biết biểu thức tiền tố và biểu thức hậu tố của biểu thức trên bằng phương pháp duyệt NLR và LRN.
- 3) Hãy tạo một heap từ dãy số sau: 8 3 5 2 20 11 30 9 18 4

Phần V. Cấu trúc tìm kiếm.

1) Cho dãy số: 8 3 5 2 20 11 30 9 18 4

Hãy vẽ cây nhị phân tìm kiếm hình thành bằng cách lần lượt thêm vào các phần tử trong dãy số trên

2) Cho cây nhị phân tìm kiếm



Hãy vẽ lại cây nhị phân tìm kiếm trên trong các trường hợp

- Sau khi xóa nút 66
- Sau khi thêm vào nút 11 và nút 55
- Sau khi xóa nút 73
- Sau khi xóa nút 25