

Môn thi: Cấu trúc dữ liệu và giải thuật

Mã lớp: IT003 – lớp chính qui đại trà, chất lượng cao

Thời gian làm bài: 75 phút

(Sinh viên không được sử dụng tài liệu)

Câu 1 :

- a. Hãy trình bày ý tưởng của giải thuật tìm kiếm tuyến tính và cho biết độ phức tạp của giải thuật. (1 điểm)

→ Đáp án tham khảo:

- Ý tưởng giải thuật (0.5 điểm):

Xuất phát từ phần tử đầu tiên (phần tử đang xét) của mảng/danh sách A có N phần tử, so sánh phần tử X cần tìm với phần tử đang xét, nếu phần tử đang xét có giá trị bằng X thì thông báo có X trong mảng/danh sách và kết thúc giải thuật, ngược lại chuyển qua xem xét với phần tử tiếp theo trong mảng/danh sách. Khi xét đến phần tử cuối mảng/danh sách nếu giá trị của phần tử vẫn không bằng X, thông báo không có X trong mảng/danh sách A, kết thúc giải thuật.

- Độ phức tạp (0.5 điểm) : $O(N)$

- b. Trình bày các bước (vẽ từng bước) giải thuật tìm kiếm tuyến tính thực hiện tìm giá trị $X=5$ trong mảng 6 số nguyên có giá trị: 81; 90; 62; 65; 12; 42 (1.5 điểm)

→ Thang điểm đề nghị : Mỗi bước thực hiện đúng : 0.25 điểm / bước.

→ Đáp án tham khảo:

- Mảng A : $N = 6$ phần tử :

| | | | | | |
|----|----|----|----|----|----|
| 81 | 90 | 62 | 65 | 12 | 42 |
|----|----|----|----|----|----|

- Bước 1 : $I = 0$ (phần tử đầu tiên trong mảng), so sánh $A[I] = A[0] = 81 \neq X = 5$, chuyển qua xem xét phần tử tiếp theo trong mảng A, $I=I+1$.
- Bước 2 : $I = 1$, so sánh $A[I] = A[1] = 90 \neq X = 5$, chuyển qua xem xét phần tử tiếp theo trong mảng A, $I=I+1$.
- Bước 3 : $I = 2$, so sánh $A[I] = A[2] = 62 \neq X = 5$, chuyển qua xem xét phần tử tiếp theo trong mảng A, $I=I+1$.
- Bước 4 : $I = 3$, so sánh $A[I] = A[3] = 65 \neq X = 5$, chuyển qua xem xét phần tử tiếp theo trong mảng A, $I=I+1$.
- Bước 5 : $I = 4$, so sánh $A[I] = A[4] = 12 \neq X = 5$, chuyển qua xem xét phần tử tiếp theo trong mảng A, $I=I+1$.
- Bước 6 : $I = 5$, so sánh $A[I] = A[5] = 42 \neq X = 5$, đã xét phần tử cuối trong mảng, thông báo không có X trong mảng, kết thúc giải thuật.

Câu 2 :

Người ta muốn lưu trữ danh sách hàng hóa tại công ty X với các thông tin chính yếu nhằm hỗ trợ nhanh trong tra cứu, với các thông tin: Tên mặt hàng (chuỗi); Giá mặt hàng (số nguyên); Số lượng còn trong kho (số nguyên). Hãy thực hiện :

- a. Định nghĩa cấu trúc dữ liệu lưu danh sách các mặt hàng theo thông tin mô tả ở trên, sử dụng cấu trúc danh sách liên kết. (1 điểm)

→ Đáp án tham khảo :

- SV có thể khai báo cấu trúc dữ liệu sử dụng danh sách liên kết đơn hoặc đôi.

→ Khai báo cấu trúc một phần tử trong danh sách : 0.5 điểm

```
typedef struct PT
{ char Ten_mat_hang [100];
  int Gia_mat_hang;
  int So_luong_kho;
  struct PT *Next;
}Node;
```

→ Khai báo cấu trúc danh sách : 0.5 điểm

```
typedef struct DS
{ node *Head;
  node *Tail;
}Danhsach;
```

- b. Viết hàm nhập vào danh sách 50 mặt hàng sử dụng cấu trúc dữ liệu ở câu 2.a, biết rằng khi nhập lần lượt từng mặt hàng sẽ thêm vào cuối danh sách. (1.5 điểm)

→ Thang điểm đề nghị :

- Viết đúng hàm nhập 50 mặt hàng : 1 điểm
- Viết đúng hàm thêm vào cuối danh sách: 0.5 điểm

→ Đáp án tham khảo:

```
void Input (Danhsach &l)
{ int i,gia,soluong;
  Node *p;
  char s[100];
  for (i=0;i<50;i++)
  { cout <<"nhap ten mat hang";
    fgets(s, 100, stdin);
    cout <<"nhap gia mat hang";
    cin >> gia;
    cout <<"nhap so luong con trong kho";
    cin >> soluong;
```

```

        p = createnode(s,gia,soluong); // SV có thể không cần viết hàm createnode
        AddTail(l,p);
    }
}

```

```

void AddTail(DanhSach &l, Node *p)
{
    if (l.Head == NULL)
        l.Head = l.Tail = p;
    else
    {
        l.Tail ->Next = p;
        l.Tail = p;
    }
}

```

- c. Viết hàm nhập vào 2 số nguyên dương x, y ($x < y$), hiển thị lên màn hình danh sách mặt hàng có số lượng trong kho lớn hơn x và nhỏ hơn y . (1 điểm)

➔ Thang điểm đề nghị :

- Nhập 2 số x, y đúng : 0.5 điểm
- Viết đúng hàm thêm in lên màn hình đúng yêu cầu: 0.5 điểm

➔ Đáp án tham khảo:

```

void Print(DanhSach l)
{
    int x,y;
    do
    {
        cout << "Nhap so nguyen duong x : ";
        cin >> x;
        if (x<=0)
            cout << "Gia tri x khong hop le, vui long nhap lai";
    }while (x<=0);
    do
    {
        cout << "Nhap so nguyen duong y : ";
        cin >> y;
        if (y<=0)
            cout << "Gia tri y khong hop le, vui long nhap lai";
    }while (y<=0);
    Node *p;
    p=l.Head;
    while (p)
    {
        if (p-> So_luong_kho >x && p-> So_luong_kho <y)
            cout << p-> Ten_mat_hang;
        p=p->Next;
    }
}

```

}

Câu 3 :

Hãy thực hiện chuyển đổi một số nguyên dương N ($N < 1000$) ở hệ thập phân sang biểu diễn ở hệ nhị phân (ví dụ : số 5 ở hệ thập phân sẽ là 101 ở hệ nhị phân), sử dụng cấu trúc ngăn xếp (stack), với các yêu cầu sau :

- a. Định nghĩa cấu trúc ngăn xếp để lưu trữ số nhị phân (1 điểm).

→ SV có thể sử dụng Mảng hoặc Danh sách liên kết để cài đặt stack.

→ Thang điểm đề nghị: SV định nghĩa cấu trúc đúng yêu cầu, trọn vẹn điểm của câu.

→ Đáp án tham khảo: sử dụng Mảng, có khai báo kích thước tối đa của mảng.

```
#define MAX 100
struct stack
{
    int n;
    int e[MAX];
};
```

→ Đáp án tham khảo: Danh sách liên kết

```
struct node
{
    int data;
    struct node *next;
};
typedef struct node *pnode;
```

- b. Viết các hàm thao tác với cấu trúc ngăn xếp trong câu 3.a : push; pop; kiểm tra stack rỗng; kiểm tra stack đầy (2 điểm).

→ Thang điểm đề nghị :

- Viết đúng hàm pop/push : 0.75 điểm
- Viết đúng hàm kiểm tra stack rỗng/đầy : 0.25 điểm

→ Đáp án tham khảo: sử dụng Mảng.

```
void init(stack &s)
{
    s.n=-1;
}
```

```
int isempty (stack s)
{
    if (s.n== -1)
        return 1;
    else return 0;
```

```

}

int isfull(stack s)
{ return (s.n==MAX-1);
}

void push(stack &s, int x)
{ if (isfull(s))
    cout <<" Ngan xep day !";
  else
  { s.n++;
    s.e[s.n]=x;
  }
}

int pop (stack &s)
{ if (isempty(s))
  { cout <<"Ngan xep rong !";
    return -1;
  }
  else
    return s.e[s.n--];
}

```

➔ Đáp án tham khảo : Danh sách liên kết. SV cần cho biết sẽ không có hàm kiểm tra stack đầy.

```

void init(pnode &h)
{ h=NULL;
}

int isempty(pnode h)
{ return (h==NULL);
}

void push (pnode &h, int x)
{ pnode p;
  p=new node;
  p->data=x;
  p->next=h;
  h=p;
}

```

```

int pop(pnode &h)
{
    int x;
    pnode p;
    x=h->data;
    p=h;
    h=h->next;
    delete p;
    return x;
}

```

- c. Viết hàm nhận đầu vào một số nguyên dương N ở hệ thập phân, chuyển đổi và hiển thị kết quả số N ở hệ nhị phân lên màn hình sử dụng cấu trúc, các hàm đã định nghĩa trong câu 3.a, 3.b (1 điểm).

➔ Thang điểm đề nghị :

- Viết đúng thao tác đưa số nhị phân vào ngăn xếp: 0.75 điểm
- Viết đúng thao tác lấy số nhị phân từ ngăn xếp, hiển thị lên màn hình : 0.25 điểm

➔ Đáp án tham khảo: sử dụng Mảng.

```

void Convert_10_2(int n)
{
    stack s;
    init(s);
    int k;
    while (n!=0)
    {
        k=n%2;
        push(s,k);
        n=n/2;
    }

    cout << "Số nhị phân là: ";
    while (isempty(s)==0)
        cout << pop(s);
}

```

➔ Đáp án tham khảo: sử dụng Danh sách liên kết

```

void Convert_10_2(int n)
{
    pnode h;
    init(h);
    int x;
    while (n!=0)
    {
        x=n%2;
        push(h,x);
    }
}

```

```
        n=n/2;
    }

    cout << "So nhi phan la: ";
    while(isempty(h)==0)
        cout << pop(h);
}
```

HẾT