

Chương 1

Tổng quan về cấu trúc dữ liệu và giải thuật

Viết chương trình hoàn chỉnh cho các bài toán sau đây

Ôn tập các kiểu dữ liệu có cấu trúc đã học

BT1-1. Cho n số nguyên dương $a_0, a_1, a_2, \dots, a_{n-1}$.

- Chèn phần tử x vào vị trí k của dãy.
- Xóa tất cả các số nguyên tố trong dãy.
- Kiểm tra dãy có tăng dần hay không ?
- Tìm số nhỏ nhất chia hết cho tất cả các số của dãy.
- Tìm các cặp số nguyên tố cùng nhau (hai số nguyên dương được gọi là nguyên tố cùng nhau nếu ước số chung lớn nhất của chúng bằng 1).
- Tìm tần số xuất hiện của các số trong dãy.

BT1-2. Cho ma trận vuông n dòng và n cột; các phần tử là các số nguyên ($0 \leq n < 100$). Viết các hàm thực hiện các yêu cầu sau:

- Tính tổng tất cả các phần tử của ma trận.
- Tìm giá trị dương nhỏ nhất của ma trận.
- Tính tổng các phần tử nằm trên đường chéo phụ.
- Kiểm tra xem các phần tử nằm trên đường chéo chính có tăng dần hay không ? (theo chiều từ góc trên bên trái xuống góc dưới bên phải).

BT1-3. Cho ma trận vuông n dòng n cột; mỗi phần tử của ma trận là một phân số (giả thiết rằng tử số và mẫu số của các phân số này là các số nguyên). Hãy thực hiện các yêu cầu sau:

- Tìm phân số có giá trị nhỏ nhất nằm trong khoảng $(0; 1)$.
- Đếm số lượng phân số nằm trong ma trận tam giác trên có giá trị nằm trong khoảng $(0, 1)$
- Sắp xếp các phân số trong ma trận tăng dần từ trái qua phải và từ trên xuống dưới.

BT1-4. Viết chương trình nhập vào chuỗi ký tự s . Hãy thực hiện các yêu cầu sau:

- Cho biết số lượng ký tự của mỗi từ.

- b. Tìm tần số xuất hiện của các ký tự trong chuỗi s (không kể ký tự khoảng trắng).
- c. Đếm xem chuỗi s có bao nhiêu từ ?
- d. Xóa n ký tự trong chuỗi kể từ vị trí thứ k.
- e. Nhập thêm chuỗi s1, hãy chèn s1 vào vị trí thứ k của s.
- f. Loại bỏ các khoảng trống dư thừa trong chuỗi

BT1-5. Cho một danh sách lưu trữ thông tin về các nhân viên trong một công ty, thông tin gồm :

- Mã nhân viên (chuỗi, tối đa là 8 ký tự)
- Họ và tên (chuỗi, tối đa là 20 ký tự)
- Năm sinh (số nguyên)
- Phòng ban (chuỗi, tối đa 10 ký tự)
- Lương cơ bản (số nguyên)
- Thưởng (số nguyên)
- Thực lãnh (số nguyên, trong đó thực lãnh = lương cơ bản + thưởng)

Hãy thực hiện các yêu cầu sau:

- a. Tính tổng thực lãnh tháng của tất cả nhân viên trong công ty.
- b. In danh sách những nhân viên có mức lương cơ bản thấp nhất.
- c. Đếm số lượng nhân viên có mức thưởng ≥ 1200000 .
- d. In danh sách các nhân viên tăng dần theo phòng ban, nếu phòng ban trùng nhau thì giảm dần theo mã nhân viên.
- e. Cập nhật tăng lương của tất cả các nhân viên lên 5%

BT1-6. Viết chương trình tạo một tập tin văn bản có tên là “DAYSO.INP” có cấu trúc như sau:

- Dòng đầu tiên ghi n (n là số nguyên dương nhập từ bàn phím).
- Trong các dòng tiếp theo ghi n số nguyên ngẫu nhiên trong phạm vi từ 1 đến 10000, mỗi dòng 10 số (các số cách nhau ít nhất một dấu cách).

Hãy thực hiện các công việc sau đây:

- a. Tìm giá trị lớn nhất của các số trong tập tin DAYSO.INP.
- b. Đếm số lượng số chẵn, số lượng số lẻ trong tập tin DAYSO.INP.
- c. Hãy đếm số lượng số nguyên tố, số chính phương, số hoàn hảo, số Armstrong trong tập tin DAYSO.INP.

Hãy ghi kết quả của các câu a,b,c trên vào tập tin văn bản có tên là “DAYSO.OUT”.

BT1-7.Viết chương trình tạo tập tin văn bản có tên là “BANGSO.INP” có cấu trúc như sau:

-Dòng đầu tiên ghi hai số m và n (m, n là các số nguyên dương nhập từ bàn phím)

-Trong m dòng tiếp theo mỗi dòng ghi n số nguyên ngẫu nhiên trong phạm vi từ 0 đến 1000 (các số cách nhau ít nhất một dấu cách)

Hãy thực hiện các công việc sau:

a.Hãy cho biết chỉ số các dòng có chứa số nguyên tố (giả thiết các dòng trong tập tin văn bản được đánh số từ 0 đến m-1).

b.Xoay vòng các cột qua phải một vị trí (cột 0 sẽ qua cột 1, cột 1 qua cột 2,... cột n-1 về cột 0).

c.Sắp xếp các phần tử tăng dần trên từng cột.

Hãy ghi các kết quả trên vào file văn bản có tên là “BANGSO.OUT”.

BT1-8. Cho mảng một chiều gồm n tọa độ điểm (giả sử hoành độ và tung độ của các điểm là các số nguyên).

a.Hãy tìm một điểm trong mảng xa gốc tọa độ nhất.

b.Hãy tìm tọa độ hai điểm gần nhau nhất.

c.Hãy xác định tọa độ của hình chữ nhật nhỏ nhất bao hết cả n điểm trên (tọa độ góc trên bên trái và tọa độ góc dưới bên phải của hình chữ nhật).

Ví dụ n = 5 và tọa độ 5 điểm là: (0,0); (0,3); (3,3); (4,1); (4,4).

Thì kết quả câu a là điểm (4,4), kết quả câu b là (3,3) và (4,4), kết quả câu c là (0,4); 4(,0).

Chọn lựa một cấu trúc dữ liệu thích hợp/độ phức tạp tính toán

BT1-9.Cho dãy n số nguyên a_0, a_1, \dots, a_{n-1} . Hãy chuyển k phần tử đầu tiên của dãy về cuối dãy.

BT1-10.Giả sử $n \geq 1$ và x là số thực. Hãy viết hàm tính giá trị của biểu thức sau đây (với độ phức tạp tuyến tính):

$$S(n, x) = \frac{x}{1} - \frac{x^2}{1 + \frac{1}{2}} + \frac{x^3}{1 + \frac{1}{2} + \frac{1}{3}} - \dots + (-1)^{n-1} \frac{x^n}{1 + \frac{1}{2} + \dots + \frac{1}{n}}$$

BT1.11. Tìm số hạng thứ n của dãy Fibonasci (giải quyết khi n là một số lớn – khi đó ta không thể sử dụng đệ quy và cũng không thể sử dụng mảng để lưu trữ).

BT1-12. Cộng hai số nguyên lớn a và b , trong đó số a có m chữ số và số b có n chữ số.

Số nguyên lớn ở đây là số có thể có đến vài trăm chữ số. Để lưu trữ các số nguyên lớn này ta có thể dùng chuỗi (mỗi ký tự của chuỗi là một chữ số) hoặc dùng mảng một chiều (mỗi phần tử của mảng một chiều là một chữ số). Tuy nhiên trong hai phương án này thì phương án dùng mảng một chiều để lưu trữ sẽ có thuật toán tốt hơn.

Gợi ý:

Algorithms:

Nhập m chữ số của số a , lưu vào mảng một chiều a .

Nhập n chữ số của số b , lưu vào mảng một chiều b .

Giả sử ta có hai số $a=97895$ và $b = 6478$

i	0	1	2	3	4
$a[i]$	9	7	8	9	5
$b[i]$	6	4	7	8	

Lưu ý nếu khi nhập mà không giống hàng bên phải thì kết quả sẽ sai, ta có thể tiến hành nhập hai số a, b theo cách sau để khắc phục tình trạng này:

Đặt \max là số lớn nhất trong hai giá trị m và n .

Việc nhập hai số a, b được tiến hành như sau:

for $i=\max-m+1; i \leq \max; i++$

$\text{cin} >> a[i];$

for $i=\max-n+1; i \leq \max; i++$

$\text{cin} >> b[i];$

Sau đó thực hiện phép cộng như cách thông thường:

$\text{remember}=0;$

for ($i=\max; i \geq 1; i--$)

{

$c[i]=(a[i]+b[i]+\text{remember})\%10;$

$\text{remember}=(a[i]+b[i]+\text{remember})/10;$

}

```

c[0]=remember;
mảng c[i] chính là tổng của hai số a và b.
Lưu ý là giá trị c[0] này chỉ xuất ra khi nó khác 0.
Đoạn chương trình xuất kết quả như sau:
if (c[0]!=0) cout<<c[0];
for (i=1;i<=max;i++)
    cout<<c[i];
    
```

Dữ liệu thử:

m =5; n = 4;

a = 97895, b = 6478

Giá trị của các phần tử của hai mảng a và b là:

a[1] = 9, a[2]=7; a[3]=8; a[4]=9. a[5]=5.
 b[2] = 6, b[3]=4; b[4]=7; b[5]=8

BT1-13. Giả sử $n \geq 0$ và x là số thực. Hãy tính giá trị của biểu thức sau đây.

$$S(n,x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Gợi ý:

Algorithms1: $O(N^2)$

```

float s=1;
for (int i=1;i<=n;i++)
    s=s+pow(x,i)/giaithua(i); // giaithua(i) = i!=1.2.3....i
    
```

Độ phức tạp theo cách này là $O(N^2)$, tuy nhiên chương trình không thể thực hiện được khi n lớn; chẳng hạn $n = 100$ - do phép tính giai thừa của n không thể thực hiện..

Algorithms2: $O(N^2)$

```

float s=1,p;
for (int i=1; i<=n;i++)
{
    p=1;
    for (int j=1; j<=i;j++)
    
```

```

        p=p*x/j;
        s=s+p;
    }

```

Độ phức tạp theo cách này vẫn là $O(N^2)$, tuy nhiên chương trình đã không cần tính giai thừa của n .

Algorithms3: $O(N)$ – độ phức tạp tuyến tính

```

float s=1,p=1;
for (int i=1;i<=n;i++)
{
    p=p*x/i;
    s=s+p;
}

```

BT1-14.

Cho dãy n số nguyên $\{a_i\}$, ở đây giả sử $i=1..n$. Dãy con liên tiếp là dãy mà thành phần của nó là các thành phần liên tiếp nhau trong $\{a\}$, ta gọi tổng của dãy con là tổng tất cả các thành phần của nó. Tìm tổng lớn nhất trong tất cả các tổng của các dãy con của $\{a\}$.

Ví dụ nếu $n = 7$;

4 -5 6 -4 2 3 -7

Thì kết quả tổng là 7.

Gợi ý:

Sau đây là một số thuật giải cho bài toán này [4]

Algorithms1: $O(N^3)$

Thuật toán đơn giản nhất có thể viết ngay là: Xét tất cả các cặp số nguyên L và U thỏa mãn $1 \leq L \leq U \leq n$; đối với mỗi cặp như vậy ta tính tổng của dãy con $a[L..U]$ và so sánh tổng này với giá trị lớn nhất hiện có:

```

for (L=1;L<=n;L++)
    for (U=L;U<=n;U++)
    {
        sum=0;
        for (int I=L;I<=U;I++)
            sum=sum+a[I];
    }

```

```

        maxsofar=max(maxsofar,sum);
    }

```

Chương trình này tuy dễ hiểu, nhưng nó chạy rất chậm. Thuật toán này có độ phức tạp là $O(n^3)$. Các bạn xem nếu $n=10000$ thì thời gian dành cho thuật toán này là quá lớn.

Algorithms2: $O(N^2)$

Ta có thể cải tiến thuật toán trên để có thuật toán với độ phức tạp là $O(n^2)$ bằng cách sử dụng hệ thức :

Tổng $a[L..U]$ = Tổng $a[L..U-1]$ + $a[U]$

```

    maxsofar=0;
    for (L=1; L<=n; L++)
    {
        sum=0;
        for (U=L; U<=n; U++)
        {
            sum=sum+a[U];
            maxsofar=max(maxsofar,sum);
        }
    }

```

Algorithms3: $O(N)$

Tổng lớn nhất trong dãy con $a[1..i]$ là tổng lớn nhất trong dãy con $a[1..i-1]$ - gọi là maxsofar hoặc tổng lớn nhất trong tất cả các tổng của các dãy con kết thúc tại i - gọi là *maxendinghere*. Chúng ta có nhận xét rằng: Dãy con lớn nhất kết thúc tại i là dãy con lớn nhất kết thúc tại vị trí $i-1$ được bổ sung thêm phần tử $a[i]$ ở cuối hoặc là dãy con rỗng trong trường hợp tổng của dãy con nhận được là số âm. Ta có thuật toán như sau:

```

    maxsofar=0;
    maxendinghere=0;
    for (i=1; i<=n; i++)
    {
        maxendinghere=max(maxendinghere+a[i],0);
        maxsofar=max(maxsofar,maxendinghere);
    }

```

}

Minh họa cho thuật toán này như sau:

i	1	2	3	4	5	6	7
A[i]	4	-5	6	-4	2	3	-7
<i>maxendinghere</i>	4	0	6	2	4	7	0
<i>maxsofar</i>	4	4	6	6	6	7	7

Thuật toán này có độ phức tạp là $O(n)$ và rõ ràng khó có thể tìm ra một thuật toán nào nhanh hơn thế.

BT1-15. Cho dãy n số nguyên a_0, a_1, \dots, a_{n-1} . Hãy tìm dãy con liên tiếp tăng dài nhất.

BT1-16. Cho dãy n số nguyên a_0, a_1, \dots, a_{n-1} . Hãy tìm đoạn con dài nhất chứa toàn số 0.

BT1-17. Cho dãy n số nguyên a_0, a_1, \dots, a_{n-1} . Hãy tìm dãy con tăng chứa nhiều số nguyên tố nhất.

BT1-18. Thực hiện phép trừ hai số nguyên lớn.

BT1-19. Thực hiện phép nhân hai số nguyên lớn.

BT1-20. Thực hiện phép chia hai số nguyên lớn.

Chương 2

Tìm kiếm & sắp xếp

Viết chương trình hoàn chỉnh cho các bài toán sau đây

BT2-1. Cho dãy số.

84 32 13 64 1 55 48

Hãy mô phỏng sắp xếp tăng dần bằng các thuật toán chọn trực tiếp, đổi chỗ trực tiếp, nổi bọt, chèn trực tiếp qua dãy số trên.

BT2-2. Cho dãy n số nguyên $a[0], a[1], \dots, a[n-1]$ đã được sắp xếp tăng dần và một số nguyên x .

a. Hãy viết hàm tìm kiếm nhị phân kiểm tra xem x có thuộc dãy số trên hay không? Nếu tìm thấy trả về giá trị i nhỏ nhất mà $a[i] = x$, nếu không trả về giá trị -1 .

b. Cho dãy $n = 8$ số nguyên như sau:

1 2 4 4 4 9 10 15

Nếu $x=4$ thì phương pháp tìm kiếm nhị phân cho ra kết quả gì?

c. Cho biết k số phần tử lớn nhất của dãy.

Ví dụ với $n=12$

9 6 2 7 9 9 6 5 7 9 6 7

Nếu $k=5$ thì kết quả là 9, 9, 9, 9, 7

BT2-3. Cho mảng 1 chiều n phần tử. Sắp xếp các số nguyên tố tăng dần, các số khác giữ nguyên giá trị và vị trí.

BT2-4. Cho mảng vuông n . Hãy tìm phần tử lớn nhất trên mỗi đường chéo song song với đường chéo chính.

BT2-5. Cho ma trận 2 chiều m dòng, n cột. Hãy sắp tăng dần các phần tử theo chiều từ trái qua phải và từ trên xuống dưới.

BT2-6. Sắp xếp các phần tử trên các đường chéo song song với đường chéo chính tăng dần.

BT2-7. Viết chương trình cho các phương pháp sắp xếp sau:

a. Đổi chỗ trực tiếp

b. Chọn trực tiếp

c.Chèn trực tiếp

d.Nổi bọt.

BT2-8.Cho dãy số.

84 32 13 64 1 55 48

Hãy mô phỏng sắp xếp tăng dần bằng các thuật toán Quick Sort, Merge Sort, Heap Sort, Shell Sort qua dãy số trên.

BT2-9.Cho mảng một chiều gồm n phần tử là các số nguyên.

Sắp xếp các số chẵn trong mảng theo thứ tự tăng, sắp xếp các số lẻ theo thứ tự giảm dần, các số 0 giữ nguyên vị trí..

BT2-10.Cho mảng một chiều gồm n phần tử là các số nguyên.

Tìm giá trị lớn nhất khác nhau của mảng

BT2-11.Cho mảng một chiều gồm n phần tử là các số nguyên.

a.Chỉ giữ lại một giá trị trong số các giá trị giống nhau.

b.Sắp xếp các số chẵn trong mảng theo thứ tự tăng, sắp xếp các số lẻ theo thứ tự giảm dần, các số 0 giữ nguyên vị trí..

BT2-12.Cho tập tin văn bản songuyen.inp chứa các số nguyên. Hãy ghi các số nguyên tố trong tập tin songuyen.inp vào tập tin nguyento.out theo thứ tự tăng dần mỗi dòng ghi 10 số, các số cách nhau ít nhất một khoảng cách.

BT2-13.Cho 2 file số nguyên được sắp tăng dần. Hãy trộn 2 file để được một file cũng được sắp tăng dần (không dùng mảng).

BT2-14.Cho một tập tin văn bản. Hãy cho biết số lượng của các số nguyên tố, chính phương, hoàn hảo, số Amstrong trong tập tin này.

BT2-15.Viết chương trình cho phương pháp sắp xếp Quick sort

BT2-16.Hãy vẽ cây phân hoạch đệ qui của thuật toán Quick-Sort trong trường hợp xấu nhất. Từ đó, chứng tỏ rằng chi phí thuật toán Quick-sort trong trường hợp này là $O(n^2)$.

BT2-17.Viết chương trình cho phương pháp sắp xếp cây (heap sort).

BT2-18.Viết chương trình cho phương pháp sắp xếp trộn trực tiếp(merge sort).

BT2-19.Viết chương trình cho phương pháp sắp xếp với độ dài bước giảm dần (shell sort).

BT2-20.Hãy cho biết số phần tử tối thiểu và tối đa trong một heap có chiều cao h ?

Chương 3

Cấu trúc danh sách liên kết

Viết chương trình hoàn chỉnh cho các bài toán sau đây

BT3-1. Cho một danh sách liên kết đơn l, mỗi nút là một số nguyên dương.

- Tìm phần tử lớn nhất danh sách l.
- Tính tổng các phần tử của danh sách l.
- Đếm xem trong danh sách l có bao nhiêu số nguyên tố ?
- Đếm xem trong danh sách có bao nhiêu số âm ? bao nhiêu số bằng 0 ? bao nhiêu số dương ?
- Đếm xem trong danh sách có bao nhiêu số bằng x ?
- Tìm phần tử dương nhỏ nhất trong danh sách.

BT3-2. Cho một danh sách liên kết đơn l, mỗi nút là một số nguyên dương.

- Xóa phần tử đầu tiên trong danh sách.
- Xóa phần tử cuối cùng trong danh sách.
- Xóa một phần tử được trỏ bởi con trỏ q.
- Xóa một phần tử ngay trước phần tử được trỏ bởi con trỏ q.
- Xóa một nút có giá trị k.

BT3-3. Cho một danh sách liên kết đơn l, mỗi nút là một số nguyên dương.

- Hãy tạo danh sách l1 chỉ chứa các số nguyên tố từ danh sách l.
- Tách danh sách l thành 2 danh sách: một danh sách chứa toàn số chẵn, một danh sách chứa toàn số lẻ.

BT3-4. Cho một danh sách liên kết đơn l, mỗi nút là một số nguyên dương.

- Trộn hai danh sách tăng dần thành một danh sách tăng dần.
- Sắp xếp các phần tử của L giảm dần theo phương pháp chọn trực tiếp

BT3-5. Viết chương trình thực hiện các yêu cầu sau:

- Khai báo cấu trúc dữ liệu của một danh sách liên kết đơn các tỉnh. Biết rằng thông tin của mỗi tỉnh bao gồm: tên tỉnh, diện tích, dân số
- Cài đặt các thao tác cơ bản cho danh sách liên kết đơn các tỉnh (thêm, sửa, xóa, duyệt).
- Tính tổng diện tích của tất cả các tỉnh trong danh sách liên kết

- d. Tìm địa chỉ của node chứa tính có diện tích lớn nhất trong danh sách liên kết.
- e. Tìm một tỉnh có dân số lớn nhất
- f. Sắp xếp danh sách tăng dần theo diện tích.

BT3-6.Viết chương trình thực hiện các yêu cầu sau:

- a. Khai báo cấu trúc dữ liệu của một danh sách liên kết đơn để lưu tọa độ các đỉnh của một đa giác lồi trong mặt phẳng OXY.
- b. Tính chu vi của đa giác.
- c. Tính diện tích của đa giác.

BT3-7. Hãy viết phần khai báo cấu trúc dữ liệu để mô tả một danh sách liên kết đơn mà mỗi phần tử chứa một số nguyên. Viết các hàm thực hiện các yêu cầu sau:

- a. Đếm các số nguyên tố của danh sách
- b. Tạo một danh sách l1 chỉ chứa các số chẵn và chia hết cho 3 từ danh sách l.
- c. Sắp xếp các phần tử của danh sách theo chiều giảm dần.

BT3-8. Hãy khai báo kiểu danh sách liên kết đơn có tên LIST mà mỗi phần tử chứa các thông tin về một sinh viên bao gồm một số nguyên dương 4 byte chỉ mã số sinh viên và một chuỗi ký tự dài tối đa 35 cho biết họ và tên của sinh viên.

Thực hiện các công việc sau:

- a. Xây dựng hàm sắp xếp danh sách sinh viên theo thứ tự tăng dần của mã số sinh viên.
- b. Xây dựng hàm thêm một sinh viên mới vào danh sách(đã có thứ tự) sao cho vẫn bảo đảm thứ tự tăng dần của mã số.

BT3-9. Cho một danh sách liên kết, mỗi nút chứa một số nguyên.

- a. Thêm một phần tử có giá trị x vào đầu danh sách
- b. Chỉ giữ lại một giá trị trong số các giá trị giống nhau.
- c. Kiểm tra xem danh sách có được sắp xếp tăng dần hay không?
- d. Đảo ngược danh sách.
- e. Sắp xếp các số chẵn trong danh sách theo thứ tự tăng, sắp xếp các số lẻ theo thứ tự giảm dần, các số 0 giữ nguyên vị trí..

BT3- 10. Hãy cho biết chi phí của mỗi công việc sau (theo O()):

- a. Tính tổng của N số tự nhiên lẻ (số nguyên dương lẻ) đầu tiên
- b. Liệt kê N phần tử của mảng

- c. Liệt kê 1 phần tử trong 1 mảng có N phần tử
- d. Tìm 1 phần tử trong một danh sách liên kết đơn có N phần tử đã được sắp thứ tự tăng dần
- e. Thêm 1 phần tử vào hàng đợi đã có sẵn N phần tử

BT3-11.a. Cài đặt thuật toán sắp xếp chèn trực tiếp trên xâu kép

- b. Cài đặt thuật toán sắp xếp chèn trực tiếp bằng cách sử dụng đệ quy.

BT3-12. Sắp xếp danh sách tăng dần thuật toán Quick sort.

BT3-13. Cộng hai đa thức (mỗi node có 3 thành phần: hệ số khác 0 của một số hạng, số mũ tương ứng và móc nối tới node tiếp theo).

BT3-14. Ta có 4 lựa chọn: 1. Stack; 2. Queue; 3. List; 4. Cả 3 CTDL này đều không thích hợp. Trong mỗi tình huống sau, hãy cho biết áp dụng lựa chọn nào là thích hợp nhất:

- a. Các khách hàng tại quầy bán vé xe lửa lấy số thứ tự để mua vé
- b. Một danh sách tên theo thứ tự ABC
- c. Các số nguyên cần phải sắp thứ tự
- d. Danh sách các món hàng đã bán trong ngày tại quầy thu ngân trong siêu thị
- e. Chương trình có sử dụng kỹ thuật Back-tracking
- f. Các máy bay đang chờ đáp xuống phi trường

BT3-15. Hãy thể hiện thuật toán sắp xếp nổi bọt trên danh sách liên kết kép.

BT3-16. Tính giá trị của đa thức $P(x)$ với x và các hệ số cho biết trước được tổ chức dưới dạng một danh sách liên kết (mỗi node có 3 thành phần: hệ số khác 0 của một số hạng, số mũ tương ứng và móc nối tới node tiếp theo).

BT3-17. Hoàn chỉnh các thao tác trên Stack và Queue.

BT3-18. Cài đặt thuật toán Quicksort theo kiểu không đệ quy.

BT3-19. Cài đặt thuật toán Mergesort trên xâu kép.

BT3-20. Tính giá trị của một biểu thức dạng chuỗi ký tự bao gồm các chữ số và các phép toán $+$, $-$, $*$, $/$, $\%$ và dấu đóng mở ngoặc đơn.

Ví dụ: $((2 + 3) * 2) - 4 / 2 = 12$

Chương 4

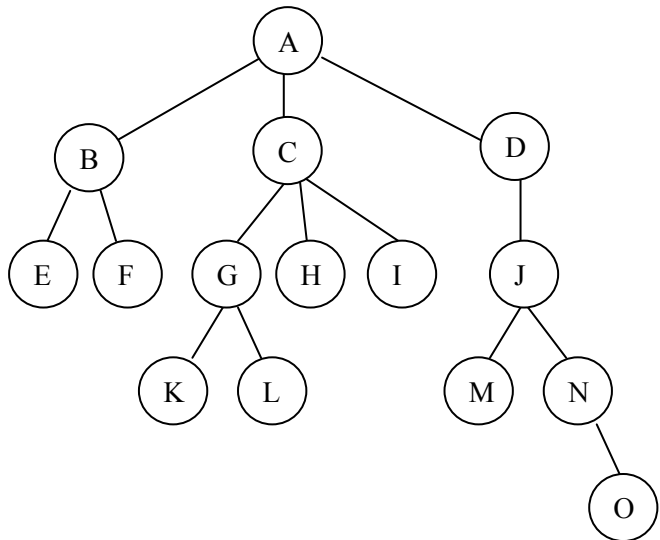
Cấu trúc cây

Viết chương trình hoàn chỉnh cho các bài toán sau đây

BT4-1.

Cho cây như hình sau. Hãy trả lời các câu hỏi sau:

- Các nút nào là nút lá ?
- Các nút nào là nút nhánh ?
- Cha của nút G là nút nào ?
- Con của nút C là các nút nào ?
- Các nút nào là anh em của nút B ?
- Mức của D, của L là bao nhiêu ?
- Bậc của B, bậc của D là bao nhiêu ?
- Bậc của cây này là bao nhiêu ?
- Chiều cao của cây này là bao nhiêu ?
- Độ dài từ A đến F, từ A đến G là bao nhiêu ?



- Có bao nhiêu đường đi từ gốc A có độ dài 3 trên cây này ?

BT4-2.

Hãy vẽ cây nhị phân tìm kiếm T biết rằng khi duyệt cây T theo thứ tự Node – Left – Right thì được dãy như sau: 9, 4, 1, 3, 8, 6, 5, 7, 10, 14, 12, 13, 16, 19.

- Hãy duyệt cây T trên theo thứ tự left – right – node, left -node-right.
- Liệt kê các nút lá của cây ? các nút nhánh của cây ?
- Hãy vẽ lại cây sau khi xoá nút 10 sao cho T vẫn là cây nhị phân tìm kiếm.

BT4-3.

- Nêu đặc điểm của cây nhị phân tìm kiếm.
- Cây nhị phân T có chiều cao là 100 thì sẽ có tối đa bao nhiêu nút ? Trong đó có tối đa bao nhiêu nút lá ? và ở mức thứ 50 của cây có tối đa bao nhiêu nút ?

BT9-4. Cho cây nhị phân tìm kiếm T gồm 12 số nguyên với phép duyệt LRN cho kết quả như sau: 1,3,2,6,7,5,4,10,9,12,11,8

- Hãy vẽ cây nhị phân tìm kiếm T.
- Duyệt cây T theo thứ tự NLR
- Vẽ lại cây sau khi xoá nút 8.

BT4-5.

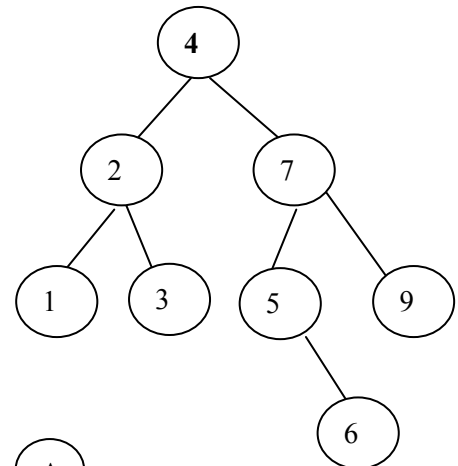
Hãy cây nhị phân tìm kiếm T gồm 11 số nguyên, với thứ tự các nút được cho như sau: 14(gốc), 11, 9, 18, 16, 20, 30, 10, 17, 1, 15

Hãy duyệt cây trên theo các thứ tự node- left-right, left- node-right, left- right- node.

Hãy cho biết chiều cao của cây T.

BT4-6.

Cho cây nhị phân T (nút gốc có giá trị là 4) như hình vẽ bên:

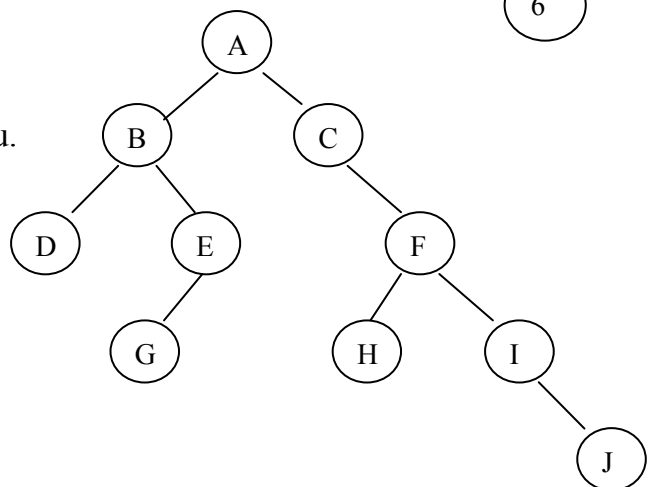


- Cây T có phải là cây nhị phân tìm kiếm không ?
- Hãy duyệt cây T theo các thứ tự node-left-right, left- node-right, left- right- node.
- Cây T có chiều cao bao nhiêu ?
- Liệt kê các nút lá của cây T.
- Liệt kê các nút nhánh của cây T.

BT4-7. Cho cây nhị phân như hình vẽ sau.

Hãy viết dãy các nút được thăm khi duyệt cây này theo

- a.thứ tự trước
- b.thứ tự giữa
- c.thứ tự sau.



BT4-8.

Tìm tất cả các cây nhị phân mà các nút sẽ xuất hiện theo một dãy giống nhau khi duyệt

- a.Theo thứ tự trước và thứ tự giữa
- b.Theo thứ tự trước và thứ tự sau
- c.Theo thứ tự giữa và thứ tự sau.

BT4-9.

Cho cây nhị phân tìm kiếm T, mỗi nút chứa một số nguyên. Hãy viết các hàm thực hiện các yêu cầu sau:

- a. Số nút lá.
- b. Số nút có đúng 1 cây con.
- c. Số nút có đúng 2 cây con.
- d. Số nút có khoá nhỏ hơn x (giả sử T là cây nhị phân tìm kiếm).
- e. Số nút có khoá nhỏ hơn x và lớn hơn y (giả sử T là cây nhị phân tìm kiếm)
- f. Đếm xem trong cây T có bao nhiêu số chẵn, bao nhiêu số lẻ ?
- g. Kiểm tra xem giá trị k có trong cây T không ?
- h. Đếm xem trong cây có bao nhiêu nút có giá trị âm? bao nhiêu nút có giá trị dương?
- i. Kiểm tra xem giá trị x có trong cây không ?
- j. Tìm giá trị lớn nhất, giá trị nhỏ nhất của cây.
- k. Tìm phần tử dương nhỏ nhất của cây.

BT4.10. Cho cây nhị phân tìm kiếm T, mỗi nút chứa một số nguyên. Hãy viết các hàm thực hiện các yêu cầu sau:

- a. Chiều cao của cây.
- b. Tìm độ lệch của cây.
- c. Xóa phần tử có giá là x trong cây
- d. Xóa các số nguyên tố ra khỏi cây
- e. Thêm một phần tử có giá trị x vào trong cây để cây vẫn là cây nhị phân tìm kiếm.

BT4-11.

Cho cây nhị phân tìm kiếm T, mỗi nút chứa một số nguyên. Hãy viết các hàm thực hiện các yêu cầu sau:

- a. Đếm số lượng nút nằm ở mức thứ k.
- b. In ra tất cả các nút ở mức thứ k của cây T
- c. In ra tất cả các nút theo thứ tự tầng 0 đến tầng thứ h-1 của cây T (h là chiều cao của cây)

BT4-12. Hãy xác định giá trị đúng/sai cho mỗi phát biểu sau:

- a. Cây nhị phân tìm kiếm luôn luôn có chiều cao $\sim \log_2 N$
- b. Chi phí tìm kiếm trung bình trên cây nhị phân tìm kiếm là $\log_2 N$

- c. Khi thêm 1 phần tử vào cây AVL, cần tối đa 1 phép xoay để điều chỉnh cây
- d. Khi xóa 1 phần tử trong cây AVL, cần tối đa 1 phép xoay để điều chỉnh cây
- e. Cây AVL luôn luôn có chiều cao $\sim \log_2 N$

BT4-13. Cho một cây nhị phân biểu diễn 1 biểu thức toán học. Biết rằng gốc của cây là Proot, mỗi nút có thuộc tính key (kiểu char), chứa 1 phép tính(+, -, *, /) hay một giá trị nguyên (các nút lá). Hãy viết hàm tính giá trị biểu thức chứa trong cây

BT4-14. Hãy nêu định nghĩa của cấu trúc dữ liệu của cây nhị phân tìm kiếm(CNPTK). Khai báo kiểu CNPTK có tên là TREE mà mỗi phần tử chứa một số nguyên và thực hiện các công việc sau:

- a. Xây dựng hàm tìm kiếm phần tử có khóa x trên CNPTK T (giá trị trả về của hàm là con trỏ trỏ đến phần tử tìm được) hoặc bằng NULL, nếu không tìm thấy.
- b. Xây dựng hàm đếm số phần tử có khóa lớn hơn k trên CNPTK T.

BT4-15. Viết chương trình quản lý một danh bạ điện thoại đơn giản bao gồm các thông tin: số thuê bao, họ tên chủ thuê bao, địa chỉ. Chương trình cho phép cập nhật: thêm/xóa/ sửa và tìm kiếm thông tin về một số thuê bao nào đó.

BT4-16. Cho một cây nhị phân T, mỗi nút là một số nguyên. Hãy viết các hàm thực hiện các yêu cầu sau:

- a. Hãy đếm số nút của cây.
- b. Cho biết chiều cao của cây AVL
- c. Kiểm tra xem T có phải là cây nhị phân tìm kiếm không ?
- d. Kiểm tra xem T có phải là cây cân bằng hoàn toàn không ?
- e. Kiểm tra xem T có phải là cây nhị phân cân bằng không ?
- f. Thêm một phần tử vào cây AVL
- g. Hủy một phần tử trên cây AVL.

BT4-17. Cho cây nhị phân T trong đó thông tin tại mỗi nút trong cây biểu diễn các thành phần thông tin của một độc giả. Biết rằng một độc giả gồm những thành phần: Mã độc giả, tên độc giả, ngày sinh, địa chỉ, ngày lập thẻ..

- a. Tìm địa chỉ của độc giả lớn tuổi nhất trong cây.
- b. Liệt kê các độc giả trong cây sinh sau năm 1975
- c. Đếm số lượng node có đủ 2 cây con có ngày lập thẻ trong ngày 09/07/2009.

d.Tìm kiếm địa chỉ theo mã độc giả.

e.Liệt kê các độc giả trong cây.

BT4-18.Đếm số nút có bậc bằng 2 trong cây nhị phân (hãy kiểm nghiệm lại rằng *số nút lá luôn bằng số nút bậc 2 cộng thêm 1*)

BT4-19.Cho một cây nhị phân tìm kiếm T. Hãy viết chương trình đếm số phần tử trong cây theo 2 cách:

a.Không dùng đệ quy

b.Có dùng đệ quy.

BT4-20.Chứng minh rằng một cây nhị phân với chiều cao h sẽ có tối đa $2^h - 1$ nút

Một số đề thi mẫu

MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (ĐỀ SỐ 1)

Thời gian làm bài: 90 phút, không kể thời gian phát đề

CÂU I (2 điểm)

Cho biết độ phức tạp của các thuật toán sau.

- a. Thêm một phần tử vào một danh sách liên kết l.
- b. Thêm một phần tử vào cây nhị phân tìm kiếm.
- c. Xóa một phần tử trong mảng 1 chiều
- d. Tìm kiếm một phần tử trong một danh sách liên kết
- e. Tìm kiếm một phần tử trong một danh sách liên cây
- f. Sắp xếp các phần tử trong danh sách liên kết
- g. Xóa một phần tử trong một danh sách liên kết
- h. Xóa một phần tử trong một cây nhị phân tìm kiếm.

Câu II (4 điểm)

Cho một danh sách liên kết đơn l khác rỗng; mỗi phần tử là một số nguyên dương.

- a. Tìm số nguyên đầu tiên trong danh sách.
- b. Tạo danh sách l1 chỉ chứa các số nguyên tố từ danh sách l.
- c. Xóa các số có giá trị lớn nhất trong danh sách l.

Câu III (4 điểm)

Cho cây nhị phân tìm kiếm T có khóa là các số nguyên.

- a. In các khóa theo chiều giảm dần.
- b. Tìm mức của nút có giá trị lớn nhất.
- c. Tìm các đường đi có độ dài là 4 trên cây.

MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (ĐỀ SỐ 2)

Thời gian làm bài: 90 phút, không kể thời gian phát đề

Câu I (2 điểm)

Cho một mảng các giá trị nguyên như sau:

Chỉ số	0	1	2	3	4	5	6	7	8	9
Giá trị	10	3	16	7	1	4	9	14	2	8

Yêu cầu:

- Cho biết kết quả sau khi biến đổi mảng trên thành dãy heap.
- Thể hiện dãy heap đó theo dạng cây nhị phân.

Câu II (4 điểm)

Cho danh sách liên kết 1. Hãy đảo ngược các phần tử của danh sách (không dùng danh sách phụ khác)

Câu III (4 điểm)

Cho cây nhị phân tìm kiếm T.

- Tìm cây con T có tổng lớn nhất (chỉ tìm 1 cây thỏa)
- Tìm giá trị nút lá ở mức sâu nhất (chỉ tìm 1 giá trị thỏa)

MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (ĐỀ SỐ 3)

Thời gian làm bài: 90 phút, không kể thời gian phát đề

Câu I (2 điểm)

Hãy vẽ cây nhị phân tìm kiếm T biết rằng khi duyệt cây T theo thứ tự left – right - node thì được dãy như sau: 1, 4, 7, 5, 3, 16, 18, 15, 29, 25, 30, 20, 8

-Hãy duyệt cây T trên theo thứ tự node - left - right.

-Cây T có chiều cao là bao nhiêu ? Tìm các đường đi từ gốc có độ dài là 4 trên cây.

Câu II(4 điểm)

Làm thế nào để chuyển các phần tử trong stack S1 sang stack S2 với những yêu cầu như sau:

- Thứ tự các phần tử trong S2 giống như trong S1
- Không được dùng thêm stack phụ, chỉ được dùng thêm vài biến phụ
- Viết giải thuật dạng mã giả.

Câu III (4 điểm)

Hãy viết hàm liệt kê tất cả các phần tử của một cây nhị phân tìm kiếm thỏa điều kiện giá trị của phần tử nằm trong khoảng [Min, Max] cho trước.

MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (ĐỀ SỐ 4)

Thời gian làm bài: 90 phút, không kể thời gian phát đề

Câu I (2 điểm)

Xét thuật giải tạo cây nhị phân tìm kiếm. Nếu thứ tự các khoá nhập vào là như sau: 8, 3, 5, 2, 20, 11, 30, 9, 18, 4 thì hình ảnh cây tạo được như thế nào ? sau đó, nếu hủy lần lượt các nút 5, 20 (theo thứ tự đó) thì cây sẽ thay đổi như thế nào trong từng bước hủy (vẽ lại cây cho mỗi bước).

Câu II (2 điểm)

Cho mảng A chứa các phần tử:

Vị trí i	0	1	2	3	4	5	6	7	8	9
A[i]	5	1	2	8	6	10	3	9	4	7

- Hãy cho biết kết quả sau khi hiệu chỉnh mảng A thành (binary) heap?
- Vẽ (binary) heap ở dạng cây nhị phân.
- Cho biết số phần tử tối thiểu và tối đa của một (binary) heap có chiều cao là h ?

Câu III (3 điểm)

- Hãy viết một hàm cài đặt thuật toán Selection-Sort bằng đệ qui.
- Cách cài đặt thuật toán này bằng đệ qui có gì tốt (không tốt) so với kiểu cài đặt bằng phép lặp thông thường.

Câu IV (3 điểm)

Cho một danh sách liên kết đơn L. Viết một hàm đảo ngược thứ tự của các phần tử trong danh sách với những yêu cầu như sau:

- Nhập (Input): danh sách L
- Xuất (Output): danh sách đảo ngược thứ tự
- Điều kiện ràng buộc: chỉ được phép duyệt danh sách L 1 lần

MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (ĐỀ SỐ 5)

Thời gian làm bài: 90 phút, không kể thời gian phát đề

Câu I (2 điểm)

Nếu ta xóa 1 phần tử khỏi cây BST và sau đó thêm nó vào lại cây. Hỏi: hình dạng của cây đó bị thay đổi không? Giải thích?

Câu II (2 điểm)

Xét thuật toán “*Sắp xếp N phần tử của mảng A*” theo mô tả sau:

Tìm phần tử nhỏ nhất trong mảng A, và copy sang mảng B tại vị trí đầu tiên, tìm phần tử nhỏ nhì trong mảng A, và copy sang mảng B tại vị trí thứ 2, thực hiện tương tự cho những phần tử còn lại của mảng A.

Yêu cầu:

- a. Viết mã lệnh(hoặc mã giải) cho thuật toán trên
- b. Cho biết chi phí của thuật toán tương ứng với trường hợp tốt nhất, xấu nhất

Câu III (3 điểm)

Cho một danh sách liên kết đơn trong đó, mỗi phần tử chứa 1 số nguyên

Hãy viết hàm xóa phần tử có giá trị lớn nhất trong danh sách.

Lưu ý:

- Chỉ được phép duyệt danh sách 1 lần
- Nếu có nhiều phần tử lớn nhất giống nhau, xóa phần tử đầu tiên

Câu IV (3 điểm)

Giả sử T là một cây nhị phân tìm kiếm mà tại mỗi nút lưu một số nguyên. Hãy khai báo kiểu dữ liệu T và xây dựng các hàm sau:

- a. Hàm thêm một node có khóa X vào T.
- b. Hàm đếm số nút có khóa lớn hơn X nhưng nhỏ hơn Y ($X < Y$).

MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (ĐỀ SỐ 6)

Thời gian làm bài: 90 phút, không kể thời gian phát đề

Câu I (2 điểm)

Hãy trình bày 2 cách chọn phần tử trung tâm (pivot) khác nhau trong thuật toán Quick-sort. Phân tích đặc điểm tốt/xấu của mỗi cách chọn dựa trên chi phí thuật toán.

Câu II (4 điểm)

Viết chương trình trên danh sách liên kết đơn mà các số là các số nguyên dương như sau:

- a. Hoán vị phần tử với phần tử đứng liên sau nó (nếu có)
- b. Đảo ngược một danh sách bằng cách thay đổi các liên kết chứ không sao chép qua một danh sách khác.
- c. Xóa tất cả các số nguyên tố trong danh sách.

Câu III (4 điểm)

Cho cây nhị phân tìm kiếm T, mỗi nút chứa một số nguyên. Hãy viết các hàm thực hiện các yêu cầu sau:

- a. In ra tất cả các nút ở mức thứ k của cây T
- b. In ra tất cả các nút theo thứ tự tầng 0 đến tầng thứ h-1 của cây T (h là chiều cao của cây)

MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (ĐỀ SỐ 7)

Thời gian làm bài: 90 phút, không kể thời gian phát đề

Câu I(2 điểm)

- a.Hãy nêu sự khác nhau của cây nhị phân tìm kiếm (BST) và Heap?
- b.Có thể liệt kê N phần tử của Heap theo thứ tự tăng dần với chi phí $O(N)$ không? Vì sao?

Câu II (4 điểm)

Hãy khai báo kiểu cấu trúc dữ liệu liên kết đơn mà mỗi phần tử chứa thông tin về một quyển sách bao gồm các trường: Mã số sách (số nguyên 4 byte), tên sách(chuỗi, tối đa 40 ký tự), tác giả (chuỗi, tối đa 30 ký tự), năm xuất bản (số nguyên 4 byte).

- a.Đếm số sách xuất bản năm X.
- b.Sắp xếp danh sách theo mã số sách.

Câu III (4 điểm)

Cho cây nhị phân tìm kiếm T, mỗi nút chứa một số nguyên. Hãy viết các hàm thực hiện các yêu cầu sau:

- a.Cho biết mức của nút có khoá nhỏ nhất trên cây T.
- b.Kiểm tra xem giá trị x có thuộc cây hay không ? Nếu có thì in đường đi từ nút gốc đến nút x.

MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (ĐỀ SỐ 8)

Thời gian làm bài: 90 phút, không kể thời gian phát đề

Câu I(2 điểm)

a.Cho dãy 15 số nguyên $a[i]$ với $i = 0..14$ như sau:

1 1 1 2 2 2 3 3 3 4 4 4 5 5 5

Cho biết vị trí trả về khi tìm kiếm giá trị $x = 4$ trong dãy trên bằng phương pháp tìm kiếm nhị phân

b.Mô phỏng các bước sắp xếp dãy số sau theo thuật toán Bubble Sort, Quick Sort: 6 2 5 1 9 7

Câu II(4 điểm)

Hãy khai báo kiểu dữ liệu danh sách liên kết đơn mà mỗi phần tử chứa thông tin về một sinh viên bao gồm các trường: mã số sinh viên (chuỗi, 6 ký tự), tên sinh viên(chuỗi, tối đa 10 ký tự), họ sinh viên (chuỗi, tối đa 30 ký tự), điểm trung bình tích lũy (số thực), loại tốt nghiệp (ký tự).

Với các kiểu dữ liệu vừa khai báo, hãy xây dựng các hàm sau:

a.Hàm xếp loại tốt nghiệp cho sinh viên. Trong đó xếp loại X(xuất sắc) nếu có điểm trung bình tích lũy (ĐTB) ≥ 9.0 ; Loại G(Giỏi) nếu có điểm trung bình tích lũy $8 \leq \text{ĐTB} < 9.0$; Loại K(khá) nếu có điểm trung bình tích lũy $7 \leq \text{ĐTB} < 8$; Loại T(trung bình) nếu có điểm trung bình tích lũy $5 \leq \text{ĐTB} < 7$.

b.Hàm sắp xếp danh sách theo điểm trung bình tích lũy.

c.Hàm sắp xếp danh sách đạt loại X, danh sách cần được sắp theo tên tăng dần, nếu tên trùng thì sắp theo họ tăng dần.

Câu III (4 điểm)

Xét thuật toán “Xóa phần tử P có 2 cây con” của cây nhị phân tìm kiếm.

Hãy viết hàm “Tìm phần tử thay thế và xóa” như sau:

`void SearchStandFor_and_Delete (NODE *P);` // P là nút cần xóa

Yêu cầu:

- Tìm “phần tử thay thế Q phần tử lớn nhất trong cây con trái” của P
- Thay P bằng Q
- Xóa Q
- Không dùng đệ qui

MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (ĐỀ SỐ 9)

Thời gian làm bài: 90 phút, không kể thời gian phát đề

Câu I(2 điểm)

Cho mảng một chiều gồm n phần tử là các số nguyên dương.

- a.Tìm giá trị lớn thứ k của mảng
- b.Chuyển các giá trị chẵn về đầu mảng, các giá trị lẻ về cuối mảng

Câu II (4 điểm)

Viết chương trình thực hiện các yêu cầu sau:

- a.Nhập danh sách liên kết kép các số nguyên.
- b.Xuất danh sách liên kết kép các số nguyên.
- c.Tính tổng các node trong danh sách liên kết kép

Câu III (4 điểm)

Giả sử T là một cây nhị phân tìm kiếm mà tại mỗi nút lưu một số nguyên. Hãy khai báo kiểu dữ liệu T và xây dựng các hàm sau:

- a.Hàm tìm kiếm một node có khóa X .
- b.Hàm đếm số nút có khóa lớn hơn X .

MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (ĐỀ SỐ 10)

Thời gian làm bài: 90 phút, không kể thời gian phát đề

CÂU I (2 điểm)

Hãy so sánh (trong giới hạn 10 dòng) những đặc tính tiêu biểu của 3 phương pháp sắp xếp: Heap sort, Merge sort và Quick sort

Câu II (4 điểm)

Cho một danh sách liên kết đơn có phần tử đầu là head; mỗi nút của nó biểu thị một điểm có tọa độ x, y trên mặt phẳng OXY. Hãy mô tả cấu trúc của danh sách trên.

Giả sử danh sách trên đã có sẵn dữ liệu. Viết chương trình tạo 2 danh sách liên kết mới h1 và h2 từ danh sách đã cho, trong đó:

h1 chứa các nút có thành phần $x > 0$ và $y > 0$.

h2 chứa các nút có thành phần $x < 0$ và $y < 0$.

Các nút còn lại của danh sách head sẽ được in ra màn hình mà không tạo xâu.

Câu III (4 điểm)

Cho một cây nhị phân T.

a. Kiểm tra xem T có phải là cây cân bằng hoàn toàn không ?

b. Thêm một phần tử vào cây AVL

c. Hủy một phần tử trên cây AVL.

Hết