

(Sinh viên không được sử dụng tài liệu)

Đề 1

MÔN: Câu trúc dữ liệu & Giải thuật

HỌ TÊN SV:	CÁN BỘ COI THI	ĐIỂM
MSSV:		
STT:		
PHÒNG THI:		

Câu 1 (0.5 điểm)

Hãy trình bày quá trình tìm $X = 4$ bằng thuật toán **tìm kiếm nhị phân** trong mảng A có thứ tự **GIẢM** dần với $N=5$ phần tử gồm $\{9, 6, 5, 3, 1\}$.

(Lưu ý mỗi bước trình bày phải có thông tin về left, mid, right)

.....

.....

.....

.....

.....

.....

.....

.....

Câu 2 (1.0 điểm)

Khi sắp xếp mảng A = {6, 8, 5, 1, 3} theo thứ tự **TĂNG** dần bằng thuật toán **Quick Sort** với pivot là phần tử ở giữa ($\text{left}+\text{right})/2$. Hãy cho biết có bao nhiêu đoạn phân hoạch và liệt kê các phần tử trong các đoạn phân hoạch đó?

(Lưu ý tính luôn ban đầu là đoạn $[0,4]$ gồm 5 phần tử 6, 8, 5, 1, 3)

Số đoạn được phân hoạch là:

Các phần tử cụ thể trong mỗi đoạn phân hoạch là:

.....
.....

Câu 3 (0.5 điểm)

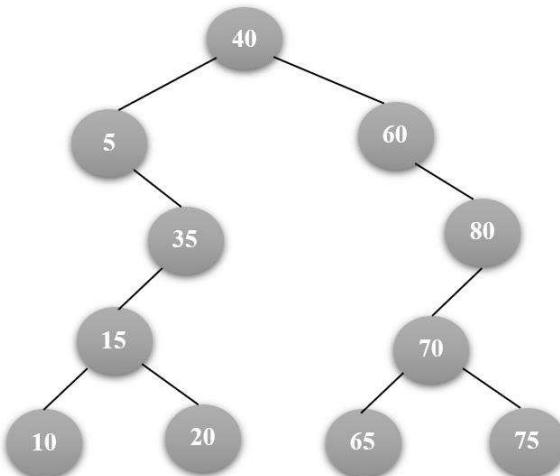
Hãy vẽ cây nhị phân T từ 2 kết quả duyệt NLR và LNR như sau:

NLR: 82, 35, 11, 63, 42, 70, 40, 14, 29

LNR: 11, 35, 42, 63, 70, 82, 40, 29, 14

Câu 4 (0.5 điểm)

Cho cây nhị phân tìm kiếm T như hình 1 bên dưới:

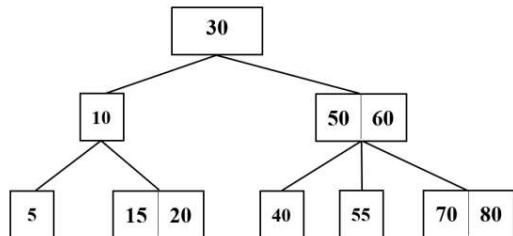


Hình 1. Cây nhị phân tìm kiếm T

Hãy vẽ cây nhị phân tìm kiếm T sau khi xóa node 40

Câu 5 (0.75 điểm)

Cho B-Tree bậc 3 như hình dưới đây.



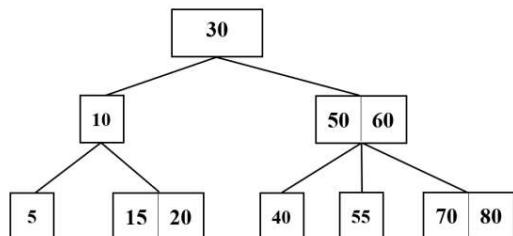
Hình 2. Cây B-Tree bậc 3

Hãy vẽ cây B-Tree sau khi thêm khóa 35 và 45

Lưu ý chỉ cần vẽ 1 cây kết quả cuối cùng

Câu 6 (0.75 điểm)

Cho B-Tree bậc 3 như hình dưới đây.



Hình 3. Cây B-Tree bậc 3

Hãy vẽ cây sau khi xóa khóa 40 (0.5đ)?

Lưu ý chỉ cần vẽ 1 cây kết quả cuối cùng

Và cho biết tên thao tác đã điều chỉnh các nút trên cây (0.25đ)?

Câu 7 (1.5 điểm)

Cho bảng băm địa chỉ mở HT, có kích thước $M=11$, hàm băm f và hàm băm lai f' lần lượt là:

$$f(key) = key \% M$$

$$f'(key, i) = \lceil f(key) + i \rceil \% M$$

Trong đó:

- Phép toán % là phép toán lấy phần dư của phép chia nguyên.
 - key là giá trị khóa.
 - i là một số nguyên cho biết lần băm lại (thăm dò) thứ i.
 - M là kích thước bảng băm.

Bảng băm HT

Chỉ số	0	1	2	3	4	5	6	7	8	9	10
Khóa	22	34	13			5	17				

a. Hãy trình bày quá trình TÌM khóa X=16 trên HT (0.75 điểm)

Lưu ý phải có kết luận cho quá trình tìm kiếm (tìm thấy hay không tìm thấy)

.....

.....

.....

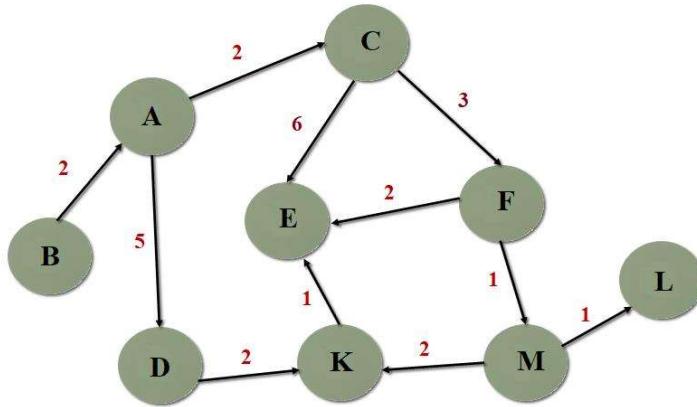
.....

.....

b. Hãy trình bày quá trình THÊM khóa X=11 trên HT (0.75 điểm)

Câu 8 (1.0 điểm)

Cho đồ thị có hướng G như bên dưới.



Hình 4. Đồ thị có hướng G

Tìm đường đi ngắn nhất từ A đến đỉnh E bằng thuật toán Dijkstra

p (đỉnh hiện hành)	Các đỉnh kề với p	Open (các đỉnh có thể xét ở bước kế tiếp)	Close (các đỉnh đã đi qua)

Câu 9 (0.5 điểm)

Cho đồ thị có hướng G ở câu 8. Tìm đường đi từ đỉnh A đến đỉnh E theo chiến lược DFS và BFS (*Lưu ý chỉ cần ghi thứ tự duyệt đỉnh*)

DFS:

BFS:

Câu 10 (2.0 điểm)

Giả sử có một **cây nhị phân tìm kiếm** T lưu trữ các số nguyên có cấu trúc như sau:

<pre>#include <iostream> using namespace std; struct Tnode { int key; Tnode* Left; Tnode* Right; }; typedef Tnode* Tree;</pre>	<pre> 9 / \ 6 16 / \ / \ 2 7 12 25 / \ / / 1 4 11 20 </pre>	<p>Ví dụ:</p> <p>Nhập x=1. Xuất ra các node tổ tiên của x là: 9, 6, 2</p> <p>Nhập x=6. Chiều cao của x là 2</p>
--	--	---

a. Hãy viết hàm xuất ra tất cả các node **tổ tiên** của node có key=x. (1.0 điểm)

b. Hãy viết hàm tính chiều cao của **node** x. (1.0 điểm)

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Câu 11 (1.0 điểm)

Cài đặt cấu trúc **QUEUE** để mã hóa 1 chuỗi nhập theo qui ước chuyển đổi các ký tự trong bảng mã ASCII và các ký tự giống nhau sẽ được gom chung như sau:

Ví dụ 1: chuỗi s="aaa" --> "973"

Giải thích kết quả "973" \rightarrow '97' là chuyển đổi của 'a' và '3' là do có 3 ký tự a liền kề nhau.

Ví dụ 2: chuỗi s="aabbb" --> "972983"

Giải thích kết quả "972983" --> gồm "972" và "983" trong đó:

"972" -> '97' là chuyển đổi của 'a' và '2' là do có 2 ký tự a liền kề nhau

("972" là mã hóa của "aa")

"983" --> '98' là chuyển đổi của 'b' và '3' là do có 3 ký tự b liền kề nhau

("983" là mã hóa của "bbb")

```
#include <iostream>
#include <string.h>
using namespace std;
// Cấu trúc của một node
struct node {
    int info;
    node* next;
};
// Cấu trúc của một Queue
struct Queue{
    node* head;
    node* tail;
};
```

```

void CreateEmptyQueue(Queue &l)
{
    l.head=l.tail=NULL;
}
node* CreateNode(int x)
{
    node* p = new node();
    if(p==NULL) exit(1);
    p->info=x;
    p->next=NULL;
    return p;
}

###Insert code here

void PrintQueue(Queue l)
{
    for(node*p=l.head; p!=NULL; p=p->next)
        cout<<p->info;
}
int main() {
    Queue L;
    CreateEmptyQueue(L);
    CreateQueue(L);
    PrintQueue(L);
    return 0;
}

```