



## VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Học phần:.....CẤU TRÚC DL &amp; TT.....Mã HP....IT3011.....Mã lớp: ..... 119183 .....

Bài thi [ ] giữa kỳ .[ X] cuối kỳ.. Năm học... 2020-2021..... Ngày thi:.....13 / 01 / 2021.....

- Không sử dụng tài liệu.
- Thời gian làm bài 90 phút.

Xác nhận của bộ môn

## PHẦN CÂU HỎI

Câu 1 (2đ) Cho hàm sau:

```
int process(int a[], int i, int j){  
    if(i > j) return 0;  
    if(i == j) return a[i];  
    int m = (i+j)/2;  
    return process(a,i,m) + process(a,m+1,j);  
}
```

- Hãy cho biết hàm **process** thực hiện công việc gì? Giải thích?
- Với giả thiết các phép toán cần thực hiện đều có thể thực hiện trong thời gian  $O(1)$ , hãy đưa ra đánh giá thời gian tính của thuật toán trong trường hợp tồi nhất theo  $O$ -lớn.

Câu 2 (1đ) Lịch trình di chuyển trong ngày của một cá nhân được lưu trữ trong danh sách liên kết theo thứ tự tăng dần thời gian, với một nút được định nghĩa như sau:

```
typedef struct aPlace{  
    int hour, min; // thời điểm ghé thăm trong ngày theo giờ và phút  
    char locName[255]; // tên địa điểm  
    struct aPlace* next; // con trỏ đến nút kế tiếp  
}PLACE;
```

Giả sử để truy vết F1 của Covid mới xuất hiện, cục y tế dự phòng phát ra thông báo những người qua một địa điểm tên **reportLocName** sau khoảng thời gian **reportHour** giờ, **reportMin** phút (cùng ngày) đều phải trình báo y tế. Hãy hoàn thiện hàm

```
int needMedicalReport(PLACE* visitedPlaces, char* reportLocName, int  
reportHour, int reportMin)
```

Hàm trả về 1 nếu cá nhân cần phải khai báo y tế, và 0 nếu ngược lại (trong đó **visitedPlaces** là con trỏ đầu đến danh sách liên kết các địa điểm cùng với thời gian ghé thăm của người cần kiểm tra).

Câu 3 (2đ)

- Cho dãy số sau cần sắp xếp theo thứ tự không tăng **7, 9, 5, 3, 1, 6, 4, 8, 2**. Hãy trình diễn thuật toán sắp xếp trộn sắp xếp dãy đã cho.

- b) Hãy lập trình ngôn ngữ C hàm `int isMinHeap(int A[], int n)` kiểm tra xem mảng `A[]` chiều dài `n` (chỉ số các phần tử từ 0 đến `n-1`) có là đồng min (min-heap) hay không ? (trả về 0 là sai, và 1 là đúng).

**Câu 4 (3đ)**

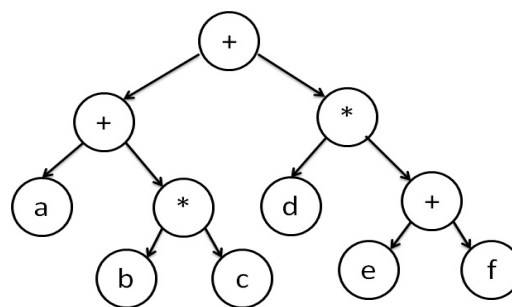
- a) Hãy vẽ cây nhị phân tìm kiếm (BST) thu được (không cần vẽ bước trung gian) bằng cách chèn liên tiếp dãy khóa sau vào BST bắt đầu từ BST rỗng: 11, 4, 8, 21, 16, 1, 3, 27, 14  
b) Hãy vẽ BST thu được sau khi loại bỏ nút có khóa bằng 11 khỏi BST thu được ở câu a)  
c) Cấu trúc dữ liệu mỗi nút của một cây nhị phân được định nghĩa như sau:

```
typedef struct TNode{
    int key; // khóa của nút
    struct TNode* left; // trỏ đến nút con trái
    struct TNode* right; // trỏ đến nút con phải
}TNode;
```

Hãy lập trình hàm `int checkBST(TNode* p)` trả về 1 nếu cây nhị phân gốc `p` là một BST và trả về 0 nếu ngược lại.

**Câu 5 (1đ)**

- a) Đưa ra kết quả duyệt cây theo thứ tự sau.  
b) Cho  $a=3, b=5, c=-4, d=6, e=2, f=9$  tính giá trị biểu thức biểu diễn bởi cây biểu thức bên.



**Câu 6 (1đ)** Cho cấu trúc mô tả vị trí ngồi của `n` người trong một phòng họp

```
typedef struct aLoc{
    double x,y; // toạ độ
    char name[51]; // tên của người họp
} LOC;
```

Danh sách được cho bởi mảng `listLoc` (các phần tử đánh số từ 0 đến `n-1`). Giả sử sau khi xét nghiệm covid có người ngồi ở vị trí tọa độ `x1,y1` bị dương tính covid, và những người trong phạm vi khoảng cách đến `(x1,y1)` nhỏ hơn hoặc bằng `d` cần được cách ly ngay. Hãy viết hàm

```
void listF1Covid(LOC *listLoc, int n, double x1, double y1, double d)
```

in ra danh sách (tên người) những người cần được cách ly (kể cả người ở vị trí tọa độ `x1,y1`).

(Chú ý: Khoảng cách tính theo Euclide)

----- Hết -----

Cán bộ coi thi không giải thích gì thêm