

## ĐỀ THI MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Học kỳ II năm học 2019-2020

Thời gian: 90 phút  
(Đề thi gồm 2 trang)

**Câu 1.** Cho pseudo-code đệ qui tính  $F(n)$  với  $n$  là số nguyên, không âm

**Function:**  $F(n)$

**Input:**  $n$

```

if  $n = 0$  then return 1;
if  $n$  lẻ then
     $y = F((n - 1)/2)$ ;
    return  $2 \cdot y \cdot y$ ;
else
     $y = F(n/2)$ ;
    return  $y \cdot y$ ;

```

- Viết thứ tự các lời gọi hàm  $F$  cần thiết để tính  $F(10)$ .
- Tính giá trị trả về của  $F(n)$  và độ phức tạp thuật toán theo  $n$ .

**Câu 2.** Các câu khẳng định sau là đúng hay sai (khi trả lời cần ghi Đúng hoặc Sai)

kèm theo giải thích ngắn gọn:

- Thuật toán quick-sort có độ phức tạp trung bình là  $O(n)$ . *Sai, ĐPT TB:  $O(N \log N)$*
- Cây (tree) có 3 đỉnh luôn là cây nhị phân khi chọn một đỉnh bất kì làm gốc. *Sai, Đ*
- Thời gian tìm đường đi ngắn nhất từ một đỉnh bằng thuật toán tìm kiếm theo chiều rộng (BFS) là  $O(|V| + |E|)$  trong đó  $|V|$  và  $|E|$  lần lượt là số đỉnh và số cạnh của đồ thị.
- Thuật toán tìm kiếm nhị phân để tìm kiếm 1 số trong 1 dãy số cho trước áp dụng được với mọi dãy số. *Sai, chỉ áp dụng vs dãy số đã đc sx*

**Câu 3.** Cho dãy số  $A = [8, 2, 12, 6, 1, 0, 9, 15, 5, 7, 3, 10]$  gồm 12 phần tử.

- Vẽ cây tìm kiếm nhị phân thu được bằng cách chèn lần các giá trị vào cây. In ra thứ tự duyệt cây tìm kiếm nhị phân vẽ được theo thứ tự in-order.
- Định nghĩa cây cân bằng là cây có độ chênh lệch chiều cao giữa cây con trái và cây con phải không quá 1 ở tất cả các đỉnh trên cây. Tìm tất cả các đỉnh mà sau khi xoá đi chỉ một đỉnh đó và thực hiện phép cập nhật lại cây, cây mới nhận được bị mất cân bằng. Giới hạn phép cập nhật lại cây khi xoá 1 đỉnh có 2 cây con là chọn đỉnh trái nhất của cây con bên phải.

**Câu 4.** Cho đồ thị vô hướng G gồm 9 đỉnh với các đỉnh được đánh số từ 1 đến 9 và danh sách cạnh như sau:

Thứ tự	1	2	3	4	5	6	7	8	9	10	11	12
Cạnh	1-6	8-9	4-6	1-4	7-3	4-5	2-7	2-8	1-8	3-9	3-1	2-3

- a. Vẽ đồ thị minh họa G. DFS
- b. Sử dụng thuật toán tìm kiếm theo chiều sâu, vẽ cây tìm kiếm xuất phát tại đỉnh 1, các đỉnh được mở rộng theo thứ tự từ bé đến lớn.
- c. Sử dụng thuật toán tìm kiếm theo chiều rộng, vẽ cây tìm kiếm xuất phát tại đỉnh 2, các đỉnh được mở rộng theo thứ tự từ bé đến lớn.

--Hết--