

ĐỀ THI GIỮA KỲ (LỚP TRỰC TUYẾN)

- Đề thi gồm 2 phần: Phần I (Phần chung): tất cả các nhóm lớp sẽ làm chung. Phần II: Dành ra cho từng nhóm lớp, từng sinh viên.
- Thời hạn: 10 ngày kể từ ngày công bố đề thi. Sẽ có điểm cộng (tùy giáo viên của từng nhóm lớp) cho những bạn làm bài và nộp bài trước. Các bài nộp sau nếu giống các bài nộp trước (đặc biệt phần 2) sẽ được chấm hội đồng, và thêm phần vấn đáp đối với sinh viên nộp sau mà giống bài.
- Nộp bài: Thư mục chứa tất cả các file liên quan
- Những yêu cầu in đậm và nghiêng, sinh viên không viết code mà trình bày trên file word,

PHẦN I: (CHUNG CHO TẤT CẢ CÁC LỚP) (6 điểm)

Câu 1: Tạo lớp cây nhị phân tìm kiếm BST. Viết chương trình với các yêu cầu sau:

1. Tạo cây nhị phân tìm kiếm với thành phần đầu vào là chuỗi các số nguyên (mỗi số nguyên trong chuỗi cách nhau một khoảng trắng).
2. Duyệt cây theo thứ tự left-right-node
3. Nhập khoá k.
 - a. Tìm xem khoá k có tồn tại trong cây BST hay không?
 - b. Nếu khoá k tồn tại trong cây thì liệt kê các khoá của các node con bên phải của node chứa khoá k này.
4. Duyệt cây theo thứ tự left-node-right.
5. Nhập khoá k.
 - a. Tìm xem khoá k có tồn tại trong cây BST hay không?
 - b. Nếu khoá k tồn tại trong cây thì xoá node chứa khoá k này. Duyệt cây sau khi xoá (Sử dụng hàm duyệt ở câu 2)
6. *Mô phỏng từng bước các câu hỏi trên (từ câu 1 đến câu 5) bằng một cây với chuỗi số thực tế (phần này trình bày bằng file word)*

Câu 2: Tạo lớp cây nhị phân tìm kiếm cân bằng AVL. Viết chương trình với các yêu cầu sau:

1. Tạo cây AVL với thành phần đầu vào là chuỗi các số nguyên (mỗi số nguyên trong chuỗi cách nhau một khoảng trắng).
2. Duyệt cây theo thứ tự left-node-right.
3. Tìm giá trị khoá lớn nhất trong cây AVL.

4. Mô phỏng từng bước các câu hỏi trên (từ câu 1 đến câu 3) bằng một cây với chuỗi số thực tế (phần này trình bày bằng file word)

PHẦN II: PHẦN MỞ RỘNG (4 điểm)

Sinh viên chọn một trong ba nội dung sau:

- a. Thuật toán tìm kiếm Breadth First Search (BFS) và Depth First Search (DFS)
 - b. Thuật toán tìm cây khung tối thiểu Minimum Spanning Tree (MST): Prim và Kruskal
 - c. Thuật toán tìm đường đi ngắn nhất Shortest way: Dijkstra và Ford-Bellman
2. Tạo lớp để xây dựng thuật toán theo một trong ba nội dung sinh viên đã chọn
 3. Chương trình chính (hàm main)
 - a. Gọi tất cả các phương thức của các lớp.
 - b. Tạo danh sách các đồ thị (có thể sử dụng mảng, ArrayList đều được), từ 3 đến 5 đồ thị
 - i. Hiển thị kết quả khi áp dụng thuật toán theo nội dung đã chọn trên danh sách các đồ thị
 - ii. So sánh kết quả sau khi áp dụng thuật toán trên các đồ thị. Hiển thị kết quả có tổng trọng số lớn nhất, nhỏ nhất giữa các đồ thị
 - c. **Mô phỏng từng bước của các thuật toán ở câu 3b bằng các đồ thị thực tế (phần này trình bày bằng file word)**

---Hết---