

BÀI ÔN TẬP CUỐI KỲ

MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

THỜI GIAN: 90 PHÚT

Quang D. C.

dungcamquang@tdtu.edu.vn

1) Cho dãy số sau:

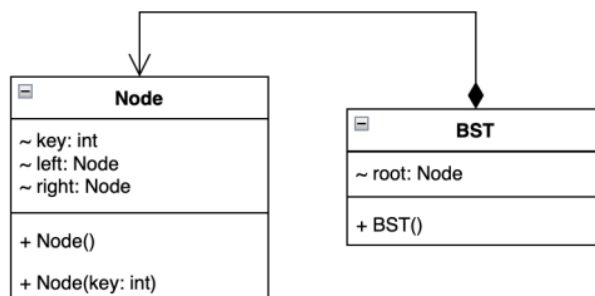
21	12	17	19	30	10	7	18	40	32
----	----	----	----	----	----	---	----	----	----

- Trình bày từng bước chạy tay giải thuật **Bubble Sort** để sắp xếp **giảm dần** dãy số trên.
 - Cài đặt thuật toán **Selection Sort** để sắp xếp **giảm dần** bằng ngôn ngữ Java.
- 2) Cài đặt các hàm sau bằng phương pháp **đệ quy** trên ngôn ngữ Java:
- public static int countVowels(String str):** trả về số lượng ký tự là nguyên âm trong chuỗi *str* truyền vào.
 - public static int findMaxSumDigitsIndex(int []arr, int n):** trả về vị trí của phần tử có tổng của các chữ số lớn nhất trong mảng *arr*. Với *n* khi gọi hàm là *arr.length - 1* và mảng luôn chứa số nguyên dương.

3) Cho dãy số sau:

20	13	15	18	10	7	5	6	16	9
----	----	----	----	----	---	---	---	----	---

- Xây dựng cây **AVL** từ dãy số trên.
- Trình bày kết quả duyệt cây theo thứ tự **NLR**.
- Lần lượt **xóa nút 20** và **xóa nút 15** (dùng successor với nút có 2 nút con), cân bằng lại cây nếu có sau mỗi lần xóa.
- Cho sơ đồ lớp:



Viết mã nguồn cài đặt bằng ngôn ngữ Java và sử dụng đệ quy (01 hàm đệ quy và 01 hàm phụ trợ) để định nghĩa các hàm sau:

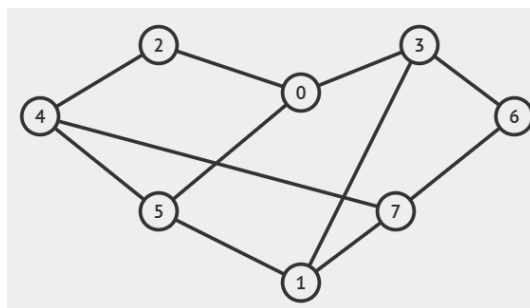
- public int sumOneChildNodes(Node x):** tính tổng giá trị của các nút có duy nhất 01 con.
- public int maxOfLevel(Node x, int level):** tìm giá trị lớn nhất trong các nút ở tầng *level* truyền vào. Root là tầng 0. Nếu *level* truyền vào lớn hơn chiều cao của cây thì trả về -1.

4) Cho dãy số sau:

13	11	20	16	27	38	22	47	8
----	----	----	----	----	----	----	----	---

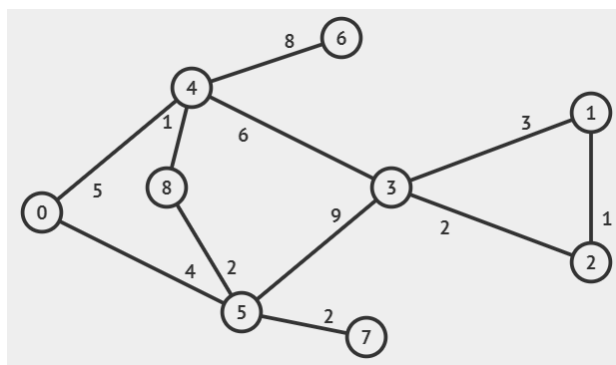
- Lần lượt thêm dãy số trên vào **Binary Max Heap**.
- Trình bày kết quả sau khi **Extract Max 2 lần liên tiếp** trên heap vừa xây dựng.

5) Cho đồ thị sau:



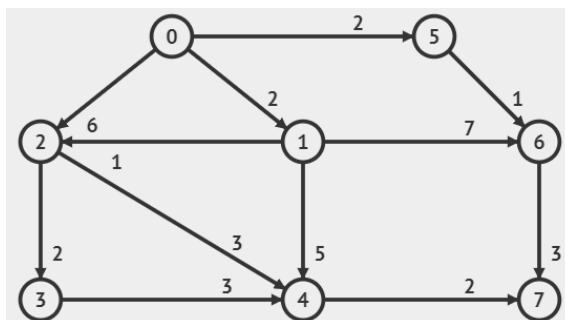
- Trình bày biểu diễn danh sách cạnh của đồ thị trên.
- Trình bày kết quả duyệt **BFS** và **DFS** của đồ thị trên, xuất phát từ đỉnh 0 và ưu tiên đỉnh có số thứ tự nhỏ.

6) Cho đồ thị sau:



Trình bày giải thuật **Prim's** tìm cây khung nhỏ nhất, xuất phát từ đỉnh **0**. Nếu 2 cạnh có cùng trọng số thì ưu tiên cạnh có số thứ tự đỉnh nhỏ nhất. Vẽ lại cây khung nhỏ nhất và tính tổng trọng số của cây khung nhỏ nhất.

7) Cho đồ thị sau:



Trình bày giải thuật **Bellman Ford's** tìm đường đi ngắn nhất từ đỉnh **1** đến các đỉnh còn lại. Vẽ đường đi, liệt kê đường đi và khoảng cách đến các đỉnh.

--HẾT--