

# Ôn tập

## Môn Cấu trúc dữ liệu và Giải thuật

### Phần 1. Danh sách (Danh sách liên kết, danh sách đặc) (4 điểm)

- Danh sách liên kết: Các thuật toán thêm, xóa, tìm kiếm, tính tổng, đếm các phần tử
- Stack/Queue: thêm phần tử, lấy phần tử ra khỏi stack/queue
- Các thuật toán sắp xếp: Selection, Bubble, insertion, Merge sort, Quick sort, radix sort: tư tưởng thuật toán, chạy từng bước thuật toán trên mảng cụ thể, đánh giá độ phức tạp của thuật toán.
- Hashing: Thêm phần tử vào hash Table, xử lý đụng độ, tìm kiếm phần tử, xóa phần tử khỏi hash table.

### Phần 2. Cây (3 điểm)

- Cây BST: Dựng cây, duyệt cây, tìm kiếm phần tử, predecessor, successor, tìm max, min, xóa phần tử
- Cây AVL: dựng cây, cân bằng cây, xóa node.
- Cây Binary Max/min Heap: Dựng cây, lấy phần tử max ra khỏi cây, Heapsort

### Phần 3. Đồ thị (3 điểm)

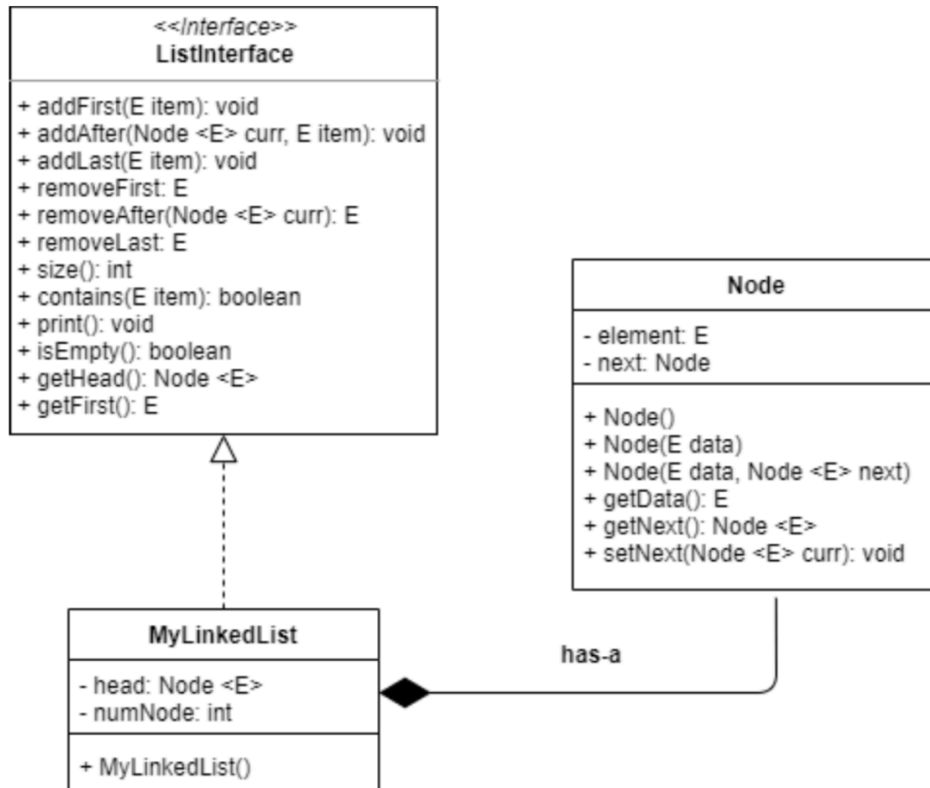
- Duyệt đồ thị: BFS, DFS
- Cây khung ngắn nhất: Prim, Kruskal
- Đường đi ngắn nhất: Dijkstra, Bellman Ford

---

### Bài tập ôn

#### Phần 1: Danh sách

1. Cho sơ đồ lớp Danh sách liên kết như hình:



Viết các phương thức sau:

- Thêm phần tử item vào đầu danh sách liên kết: `void addFirst(E item)`
  - Thêm phần tử item vào sau node hiện hành: `void addAfter(Node<E> curr, E item)`
  - Thêm phần tử item vào cuối danh sách liên kết: `void addLast(E item)`
  - Xoá phần tử đầu tiên: `void removeFirst()`
  - Xoá phần tử sau node hiện hành: `void removeAfter(Node<E> curr)`
  - Xoá phần tử cuối cùng: `void removeLast()`
  - Tìm kiếm phần tử item đầu tiên tồn tại trong DSLK. Nếu có thì trả về Node chứa item, nếu không có trả về null
2. Sử dụng sơ đồ lớp như câu 1 để cài đặt cấu trúc dữ liệu Stack và Queue. Viết các phương thức sau
- Thêm phần tử item vào Stack
  - Lấy phần tử ra khỏi Stack
  - Thêm phần tử vào Queue
  - Lấy phần tử ra khỏi Queue
3. Sắp xếp:
- Cho biết ý tưởng của các thuật toán sắp xếp: Selection, Bubble, insertion, Merge sort, Quick sort, radix sort
  - Cho mảng các số nguyên: 8 10 1 6 5 20 30 15. Chạy từng bước 6 thuật toán trên cho mảng này
  - Đánh giá độ phức tạp cho 6 thuật toán trên.
4. Hashing:
- Cho hash method:  $\text{hash}(k) = k \% m$ . Cho hash table có kích thước  $m = 7$ .

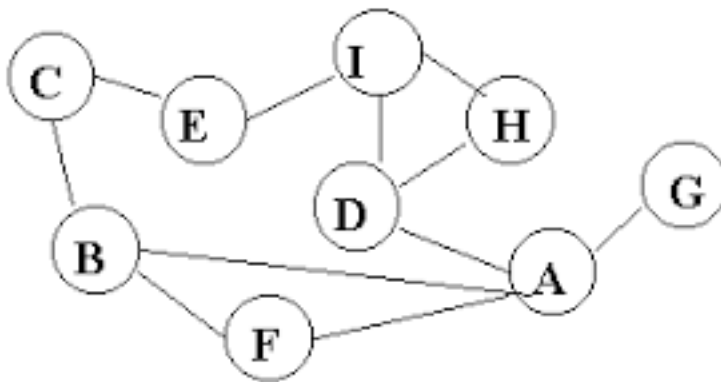
- Lần lượt thêm các phần tử sau vào hash table: 8 10 1 6 5 20 15. Nếu xảy ra đụng độ hãy sử dụng phương pháp Linear Probing để giải quyết đụng độ.
- Cho biết có bao nhiêu phép so sánh khi tìm kiếm phần tử 15 trên hash table.
- Lần lượt xoá các phần tử 1, 6, 15. Thêm phần tử 22, 27, 29.
- Cho biết có bao nhiêu phép so sánh khi tìm kiếm phần tử 27 trên hash table.

## Phần 2: cây

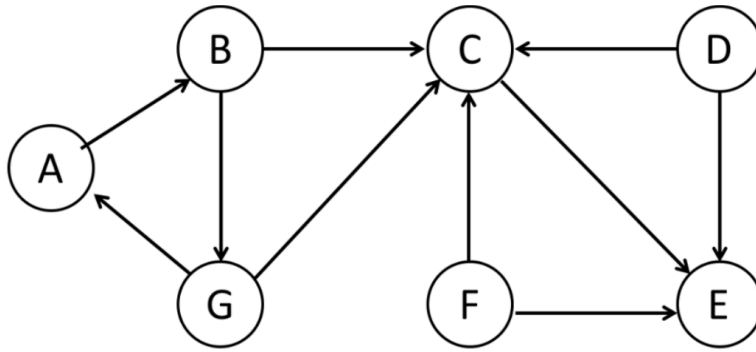
1. Cây BST: Cho dãy các khoá: 20 30 15 10 40 5 22 35 2 45
  - Dựng cây BST với các khoá trên
  - Chạy từng bước duyệt cây BST với các khoá trên theo các thuật toán: LNR, LRN, NLR, RNL, RLN, NRL
  - Tìm predecessor, successor cho các khoá: 2, 45, 20.
  - Viết thuật toán cho các bài toán sau: tìm max, min, tính tổng cây BST; tính tổng cây con bên trái, bên phải của Node x.
  - Xoá các node: 20, 10, 5.
2. Cây AVL: Cho dãy các khoá: 20 30 40 15 25 23 5 22 35 2
  - Dựng cây AVL với các khoá trên
  - Chạy từng bước duyệt cây AVL với các khoá trên theo các thuật toán: LNR, LRN, NLR, RNL, RLN, NRL
  - Tìm predecessor, successor cho các khoá: 2, 45, 20.
  - Lần lượt xoá các node: 2, 5, 15.
3. Cây Binary Max Heap: Cho dãy các khoá: 20 30 40 15 25 23 5 22 35 2
  - Dựng cây Binary max heap
  - Thực hiện dựng lại cây khi lấy các phần tử max ra khỏi cây.
  - Thực hiện từng bước sắp tăng dần dựa trên cây binary max heap.

## Phần 3: Đồ thị

1. Duyệt đồ thị bằng thuật toán BFS, DFS:
  - a. Đỉnh gốc là: A

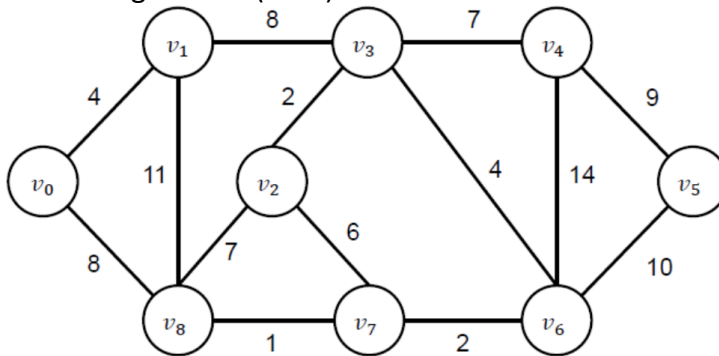


- b. Đỉnh gốc là: A

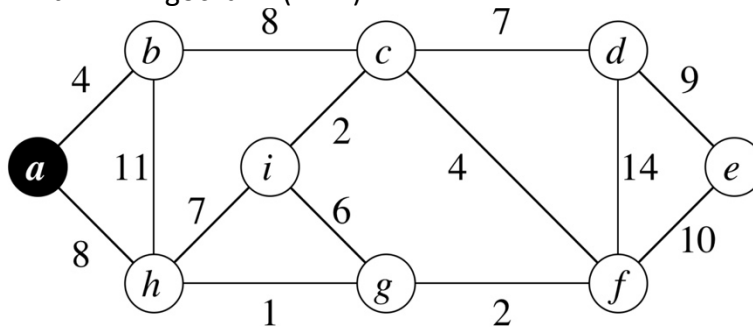


2. Tìm cây khung nhỏ nhất bằng thuật toán Prim và Kruskal cho các đồ thị sau:

a. Đỉnh gốc là:  $v_0$  (Prim)

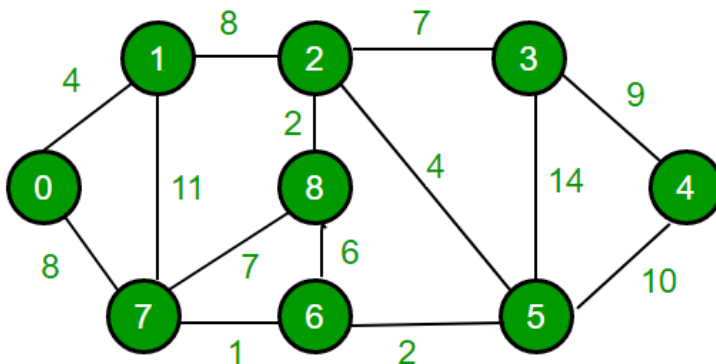


b. Đỉnh gốc là: A (Prim)

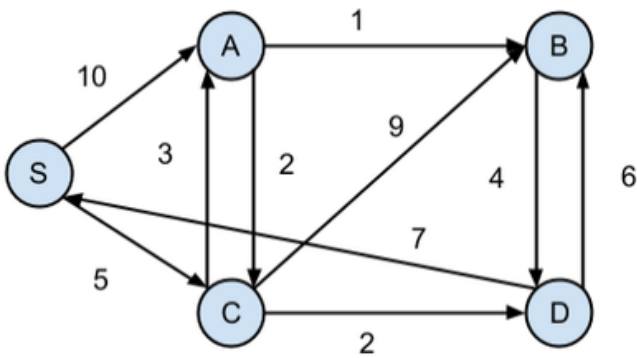


3. Sử dụng thuật toán Dijkstra để tìm đường đi ngắn nhất cho các đồ thị sau:

a. Đỉnh gốc là 0

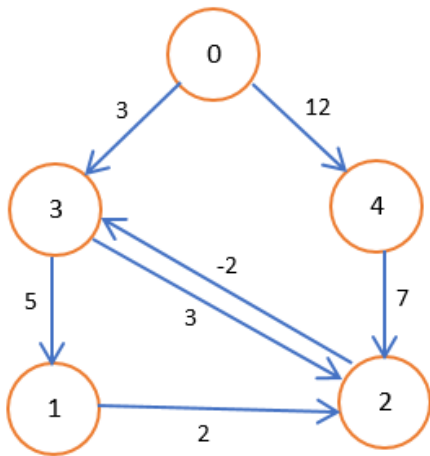


b. Đỉnh gốc là S



4. Sử dụng thuật toán Bellman Ford để tìm đường đi ngắn nhất cho các đồ thị sau:

a. Đỉnh gốc là 0



b. Đỉnh gốc là S

