**NGĂN XẾP (STACK)**

1. **Khai báo**

import java.util.\*;

Stack <E> st = new Stack<>();

Các hàm cơ bản:

* push
* pop
* peek: xem giá trị phần tử đầu tiên
* empty

1. **Ý nghĩa trong lập trình**

* Giúp quản lý và truy vết cặp dấu ngoặc (hoặc các cấu trúc có mở và đóng).

Gặp dấu mở (thẻ mở) thì push. Gặp dấu đóng (thẻ đóng) thì pop

* Giúp chuyển đổi qua lại các cách biểu diễn phép toán: tiền tố, trung tố, hậu tố (bản chất cũng dựa trên kiểm soát cặp dấu ngoặc, hoặc độ ưu tiên phép toán)
* Kiểm soát các giá trị đầu tiên nhỏ hơn hoặc lớn hơn tiếp theo trong dãy số.

(Left Right Array). Khi dùng stack độ phức tạp bài toán về cỡ O(n).

* Kiểm soát quá trình mã hóa và giải mã theo quy tắc.

1. **Áp dụng vào các nhóm bài toán.**

* Bài toán Kiểm tra dãy ngoặc đúng
* Bài toán Kiểm tra biểu thức toán học
* Bài toán Hình chữ nhật có diện tích lớn nhất

**HÀNG ĐỢI**

1. **Khai báo**

Queue <E> q = new LinkedList<>();

add

poll

offer

isEmpty()

1. **Ý nghĩa**

Ý nghĩa quan trọng nhất là để duyệt theo chiều rộng

* Từ trạng thái xuất phát, mỗi bước loang đến tất cả các trạng thái có thể đến được. Cho đến khi gặp trạng thái đích.
* Mỗi bước loại bỏ một trạng thái ra khỏi queue và cho tất cả các trạng thái đến được từ nó vào.
* Duyệt chiều rộng giúp chúng ta tìm ra cách chuyển trạng thái với ít bước nhất.

1. **Các nhóm bài toán áp dụng**

* **Nhóm 1:** BFS nhị phân
  + - * Các trạng thái không cần đánh dấu. Vì các trạng thái không thể trùng nhau.
* **Nhóm 2:** BFS đánh dấu bằng mảng
  + - * Sử dụng mảng một chiều hoặc hai chiều để đánh dấu các trạng thái đã đi qua.
* **Nhóm 3:** BFS đánh dấu bằng set hoặc map