

# Weekly Homework 5 Report

## 1 Stack

Cách thức hoạt động của stack diễn ra như sau:

1. Một con trỏ, có tên gọi là TOP, được sử dụng để theo dõi phần tử trên cùng trong ngăn xếp.
2. Khi khởi tạo ngăn xếp, ta sẽ đặt giá trị của nó là -1 để có thể kiểm tra xem ngăn xếp có trống rỗng không bằng cách so sánh giá trị của TOP với giá trị là -1.
3. Khi đẩy một phần tử vào, ta sẽ kiểm tra xem ngăn xếp đã đầy hay chưa (đầy hay không tùy thuộc vào số lượng phần tử mà Stack có thể chứa do người lập trình định nghĩa), nếu chưa thì tăng giá trị của TOP lên 1 đơn vị và đặt phần tử mới vào vị trí mà TOP trỏ đến. Nếu đầy thì đưa ra thông báo là không thể thêm phần tử nữa.
4. Khi lấy ra một phần tử, ta sẽ kiểm tra xem ngăn xếp có rỗng hay không, nếu không thì trả về phần tử được trỏ đến bởi TOP và giảm giá trị của nó đi 1 đơn vị. Nếu có thì đưa ra thông báo là ngăn xếp rỗng.
5. Trước khi đẩy phần tử vào, ta sẽ kiểm tra xem ngăn xếp đã đầy hay chưa
6. Trước khi lấy ra phần tử, ta sẽ kiểm tra xem ngăn xếp rỗng hay không

## 2 Queue

Cách thức hoạt động của Queue diễn ra như sau:

Một con trỏ, có tên gọi là HEAD và TAIL, được sử dụng để theo dõi phần tử đầu và cuối trong hàng đợi (queue).

1. Khi khởi tạo hàng đợi (queue), ta sẽ đặt giá trị của HEAD và TAIL là -1 để có thể kiểm tra xem hàng đợi có trống rỗng không. Điều này thực hiện bằng cách so sánh giá trị của HEAD và TAIL với giá trị -1. Nếu cả hai giá trị này đều là -1, hàng đợi được xem là trống.
2. Khi thêm một phần tử vào hàng đợi (enqueue), ta sẽ kiểm tra xem hàng đợi đã đầy hay chưa (đầy hay không tùy thuộc vào số lượng phần tử mà queue có thể chứa, điều này được xác định bởi người lập trình khi khai báo hàng đợi). Nếu hàng đợi chưa đầy, ta sẽ tăng giá trị của TAIL lên 1 đơn vị và đặt phần tử mới vào vị trí mà TAIL trỏ đến. Nếu hàng đợi đã đầy, sẽ có thông báo cho người dùng rằng không thể thêm phần tử nữa.
3. Khi lấy ra một phần tử từ hàng đợi (dequeue), ta sẽ kiểm tra xem hàng đợi có rỗng hay không. Nếu hàng đợi không rỗng, ta sẽ trả về phần tử được trỏ đến bởi HEAD và sau đó tăng giá trị của HEAD lên 1 đơn vị. Nếu hàng đợi rỗng, sẽ có thông báo cho người dùng rằng không thể lấy phần tử vì hàng đợi trống.

4. Trước khi thêm phần tử vào (enqueue), ta sẽ kiểm tra xem hàng đợi đã đầy hay chưa để đảm bảo không vượt quá kích thước hàng đợi đã được định nghĩa.
5. Trước khi lấy phần tử ra (dequeue), ta sẽ kiểm tra xem hàng đợi có rỗng hay không để tránh việc lấy phần tử khi không có dữ liệu trong hàng đợi.

### 3 Git/Github

