

BÀI BÁO CÁO THUẬT TOÁN DSA TUẦN 5

Nguyễn Đình Trung Kiên - 24120195

Ngày 28 tháng 4 năm 2025

1 STACK

1.1 initializeStack:

- Khởi tạo một Stack rỗng và trả về con trỏ tới Stack rỗng đó.
- Gán giá trị NULL cho Top của STACK để các thao tác sau với STACK đều hợp lệ.

1.2 stackPush:

- Tương tự như thao tác chèn một NODE mới vào đầu Linked List.
- Thao tác chèn NODE với vào đỉnh STACK ở cả hai trường hợp là STACK rỗng và không rỗng là như nhau (đều phải cập nhật lại Top của STACK cho NODE mới).
- Tuy nhiên, ở trường hợp STACK không rỗng thì ta cần nối Top mới vào đầu STACK để hoàn thành.

1.3 stackPop:

- Tương tự như thao tác xóa đi NODE đầu của Linked List, tuy nhiên cần thêm thao tác trả về giá trị của NODE bị xóa đó.
- Lưu lại địa chỉ tới NODE Top cần xóa cùng giá trị mà NODE đó lưu trữ để giải phóng và trả về.
- Cập nhật lại NODE Top đến NODE tiếp theo của nó (sẽ cập nhật luôn về rỗng nếu STACK chỉ có đúng một NODE).

- Giải phóng NODE Top cũ và trả về giá trị đã lưu.

1.4 size:

- Dùng một biến con trỏ để duyệt hết STACK từ Top và một biến đếm cập nhật liên tục qua mỗi NODE được duyệt.
- Trả về biến đếm đó sau khi duyệt hết và đó cũng là số lượng NODE trong STACK.

1.5 isEmpty:

- Trả về phủ định của phép kiểm tra Top của STACK có hợp lệ hay không.
- Từ phép phủ định trên, hàm sẽ trả về True nếu STACK rỗng (khi STACK rỗng thì việc phủ định phép kiểm tra đó sẽ trả về True) và ngược lại.

1.6 Hàm hỗ trợ:

- createNode: Hỗ trợ tạo một NODE mới từ một số nguyên được cung cấp.
- printStack: In tất cả NODE của STACK.
- freeStack: Giải phóng bộ nhớ STACK theo yêu cầu.

2 QUEUE

2.1 initializeQueue:

- Khởi tạo một QUEUE mới, gán NULL cho cả Head và Tail của QUEUE để thực hiện các thao tác khác một cách hợp lệ.
- Trả về con trỏ tới QUEUE vừa tạo.

2.2 enqueue:

- Tương tự như thao tác chèn một NODE mới vào đuôi Linked List.
- Tạo NODE mới với giá trị được truyền vào hàm.
- Nếu QUEUE rỗng thì cho cả Head và Tail trỏ vào NODE mới đó.
- Ngược lại chỉ cần nối Tail của QUEUE vào NODE mới và cập nhật lại.

2.3 dequeue:

- Tương tự thao tác xóa một NODE ở đầu Linked List.
- Nếu QUEUE rỗng, trả về -1.
- Lưu lại con trỏ với Head cần xóa và giá trị nó lưu trữ.
- Ta cập nhật lại Head của QUEUE đến phần tử tiếp theo của nó.
- Nếu sau đó QUEUE rỗng, ta cần cập nhật lại cả Tail của QUEUE.
- Giải phóng Head cũ và trả về giá trị đã được lưu lại.

2.4 size:

- Dùng một con trỏ duyệt cả QUEUE và một biến đếm như phần size ở STACK.
- Trả về số lượng NODE sau khi duyệt hết QUEUE.

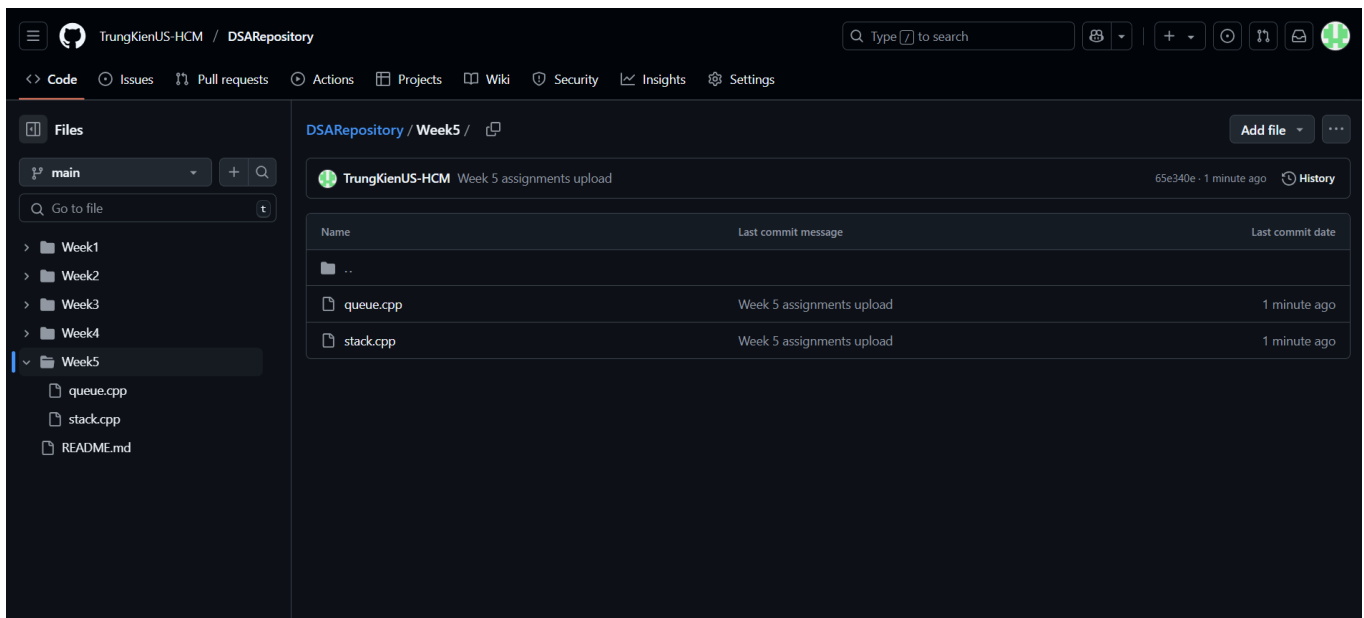
2.5 isEmpty:

- Trả về phủ định của phép kiểm tra NODE Head của QUEUE có bằng NULL hay không.
- Khi NODE Head bằng NULL thì hàm trả về True (nghĩa là QUEUE đang rỗng) và ngược lại.

2.6 Hàm hỗ trợ:

- createNode: Tạo NODE mới từ số nguyên được cung cấp.
- printQueue: In tất cả các NODE của QUEUE.
- freeQueue: Giải phóng bộ nhớ của QUEUE.

3 Hình ảnh minh chứng việc up file bài tập lên Github:



Hình 1: Minh chứng