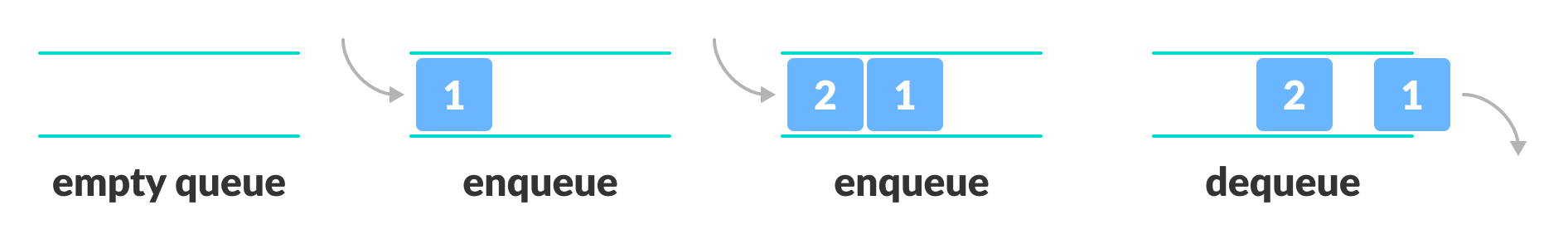
1. **QUEUE (Hàng đợi 2 đầu)**
2. **Queue là gì**

* Tương tự như Stack, Queue (hàng đợi) cũng là một cấu trúc dữ liệu. Queue tuân theo quy tắc First in first out(nhập trước xuất trước) - FIFO:
* Theo thuật ngữ lập trình, việc đưa các mục vào hàng đợi được gọi là **enqueue** và việc xóa các mục khỏi hàng đợi được gọi là **dequeue** .



1. **Các thao tác trên Queue**

* **EnQueue**: Thêm phần tử vào cuối*(rear)* của Queue.
* **DeQueue**: Xóa phần tử khỏi đầu*(front)*của Queue. Nếu Queue rỗng thì thông báo lỗi.
* **IsEmpty**: Kiểm tra Queue rỗng.
* **Front**: Lấy giá trị của phần tử ở đầu*(front)* của Queue. Lấy giá trị không làm thay đổi Queue.

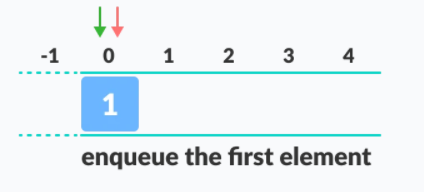
1. **Cách hoạt động của Queue**

* Hai con trỏ **FRONT** và **REAR**
* **FRONT** theo dõi phần tử đầu tiên của hàng đợi
* **REAR** theo dõi phần tử cuối cùng của hàng đợi
* Ban đầu, đặt giá trị của **FRONT** và **REAR** là -1

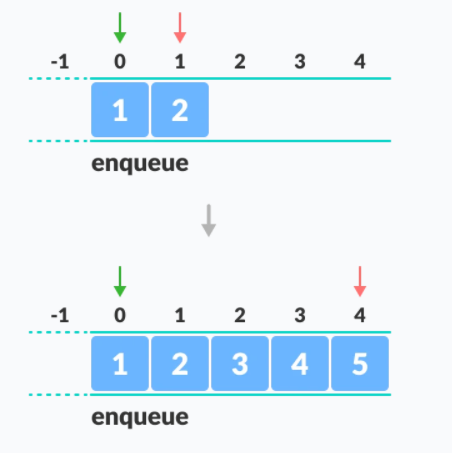


1. **Hoạt động Enqueue**

* Kiểm tra xem hàng đợi đã đầy chưa
* Với phần tử đầu tiên, đặt giá trị của **FRONT** đến 0
* Tăng chỉ số của **REAR** lên 1(**REAR** là chỉ số của phần tử sẽ được thêm ở lần tiếp theo. Do đó, thêm xong rồi ta mới tăng rear lên 1 đơn vị.)



* Thêm phần tử mới vào vị trí được trỏ bởi **REAR**

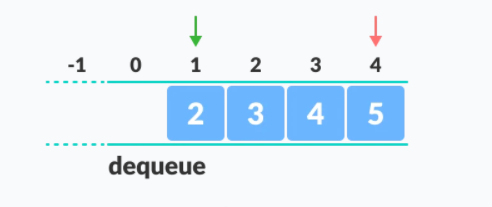


1. **Hoạt động Dequeue**

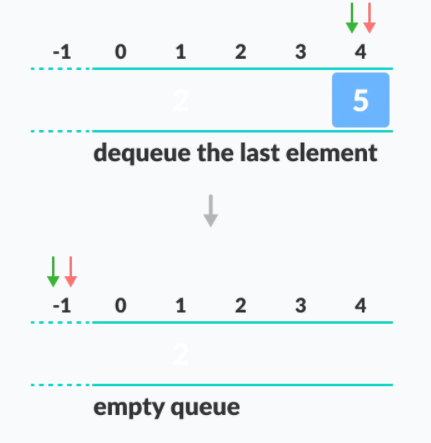
* Kiểm tra xem hàng đợi có trống không
* Trả về giá trị được trỏ bởi **FRONT**



* tăng chỉ số của **FRONT** lên 1



* Đối với phần tử cuối cùng, đặt lại các giá trị của **FRONT** và **REAR** là -1



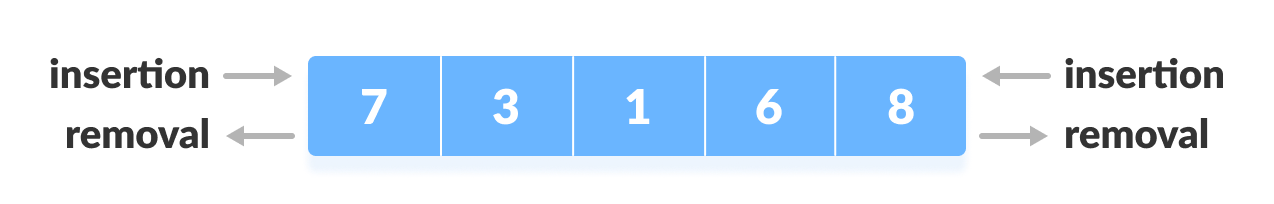
1. **Cài đặt**
2. **Độ phức tạp**

* Độ phức tạp của các hoạt động enqueue và dequeue trong queue sử dụng một mảng là O(1)

**II. DEQUE**

1. **Deque là gì**

* Deque hoặc Double Ended Queue là một loại hàng đợi trong đó việc chèn và loại bỏ các phần tử có thể được thực hiện từ phía trước hoặc phía sau. Do đó, nó không tuân theo quy tắc FIFO (First In First Out)



1. **Các thao tác trên Deque**

- push\_front(x): Đẩy x vào đầu deque

- push\_back(x): Đẩy x vào cuối deque

- pop\_front(): Loại bỏ phần tử ở đầu deque

- pop\_back(): Loại bỏ phần tử ở cuối deque

- empty(): Kiểm tra Deque có rỗng không

- size(): Trả về số phần tử đang có trong deque

**4. Cài đặt**

- Để sử dụng Deque, chúng ta có thể sử dụng các thư viện hỗ trợ có sẵn của C++ (#include<deque> )

- Cú pháp

deque <object\_type> deque\_name;

object\_type: kiểu dữ liệu (int, float, struct, …)

deque\_name: tên biến deque