

Data Structures and Algorithms

HUS HKI, 23 - 24

Assignment 4

Lecturer: Vũ Đức Minh - Trần Bá Tuấn

§ Queue - Hàng đợi §

Phần 1: Mục tiêu

- Sinh viên nắm được định nghĩa, các thao tác, và độ phức tạp của các thao tác ứng với cấu trúc dữ liệu hàng đợi (Queue)
- Sinh viên triển khai cấu trúc dữ liệu hàng đợi trên mảng và danh sách liên kết.

Phần 2: Thực hành

(1) Bài tập

Bài tập 1. *Xây dựng giao diện QueueInterface dưới đây.*

```
1 public interface QueueInterface<E> extends Iterable<E> {  
2     public void enqueue(E element);  
3     public E dequeue();  
4     public boolean isEmpty();  
5 }
```

Xây dựng kiểu dữ liệu Queue sử dụng mảng với gợi ý dưới đây.

```
1 public class ArrayQueue<E> implements QueueInterface<E> {  
2  
3     private E[] queue;  
4     private int n = 0;  
5     private int top = 0;  
6     private int count = 0;  
7     private int default_size = 100;  
8  
9     public ArrayQueue(int capacity) {  
10         n = capacity;  
11         queue = (E[]) new Object[capacity];  
12     }  
13  
14     public ArrayQueue() {  
15         n = default_size;  
16         queue = (E[]) new Object[default_size];  
17     }  
18  
19     @Override  
20     public void enqueue(E element) {  
21         // TODO Auto-generated method stub  
22     }
```

```

23
24     @Override
25     public E dequeue() {
26         // TODO Auto-generated method stub
27         return null;
28     }
29
30     @Override
31     public boolean isEmpty() {
32         // TODO Auto-generated method stub
33         return false;
34     }
35
36     @Override
37     public Iterator<E> iterator() {
38         // TODO Auto-generated method stub
39         return new ArrayQueueIterator();
40     }
41
42     class ArrayQueueIterator implements Iterator<T> {
43         private int current = top;
44         private int num = 0;
45         @Override
46         public boolean hasNext() {
47             // TODO Auto-generated method stub
48             return num < count;
49         }
50
51         @Override
52         public E next() {
53             // TODO Auto-generated method stub
54             E data = queue[(current + num) % n];
55             num++;
56             return data;
57         }
58     }
59 }

```

Yêu cầu:

- Hoàn thiện các thân hàm ở trên.
- Xây dựng kiểu dữ liệu Queue sử dụng danh sách liên kết.
- Viết các file để kiểm tra sự hoạt động của chương trình.

Bài tập 2. Sử dụng queue kết hợp với stack đã xây dựng để viết chương trình kiểm tra chuỗi Palindrome.

(2) Bài tập thêm

Bài tập 3. Triển khai Queue bằng danh sách liên kết

<https://practice.geeksforgeeks.org/problems/implement-queue-using-linked-list/1>

- Nguồn: <https://www.geeksforgeeks.org/queue-linked-list-implementation/>

Bài tập 4. Triển khai queue bằng 2 stacks

<https://practice.geeksforgeeks.org/problems/queue-using-two-stacks/1>

- Mục đích là kiểm tra kỹ năng triển khai các thao tác push/pop của stack thông qua các thao tác của hàng đợi.
- Độ phức tạp của các thao tác stack trong trường hợp này không là $O(1)$ cho cả hai phép toán mà là $O(N)$ và $O(1)$.

Bài tập 5. Số cuộc gọi gần đây

<https://leetcode.com/problems/number-of-recent-calls/>

- Trả về số cuộc gọi trong 3001s tính cả thời điểm hiện tại.
- Khi thực hiện thao tác ping, thêm giá trị t vào cuối danh sách.
- Trả về số phần tử trong danh sách có giá trị nằm trong khoảng $[t - 3000, t]$.
- Sử dụng cấu trúc dữ liệu queue để thực hiện các thao tác trên: ping - thêm 1 phần tử t vào queue, sau đó xóa các phần tử khỏi queue nếu giá trị của nó nhỏ hơn $t - 3000$. Kích thước của queue sau khi thực hiện phép xóa chính là đáp án cần tìm.

Bài tập 6. Triển khai queue vòng

<https://leetcode.com/problems/design-circular-queue/>

- Nguồn: <https://www.geeksforgeeks.org/circular-queue-set-1-introduction-array-implementation/>
- Nguồn khác: <https://www.geeksforgeeks.org/circular-queue-set-2-circular-linked-list-implementation/?ref=rp>

Bài tập 7. Triển khai deque vòng

<https://leetcode.com/problems/design-circular-deque/>

- Nguồn: <https://www.geeksforgeeks.org/implementation-deque-using-circular-array/>
- Nguồn tham khảo thêm: <https://www.geeksforgeeks.org/implementation-deque-using-doubly-linked-list/>