Queue

First in, first out → FIFO

Syntax

Interface

Class

Queue < Integer > que = new ArrayDeque <>();

functions

1) add → que.add(5); , que.offer(7);

2) remove → que.remove();

size → que.size();

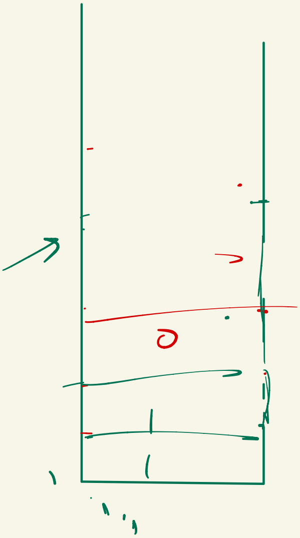
3) first → que.peek();

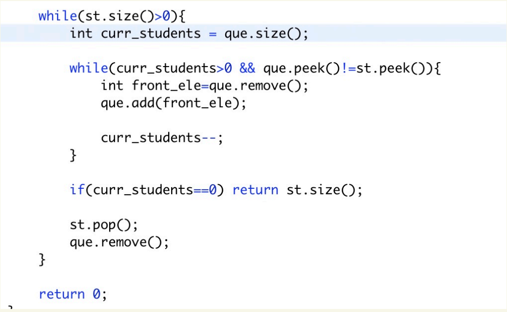
0 → circular

1 → square

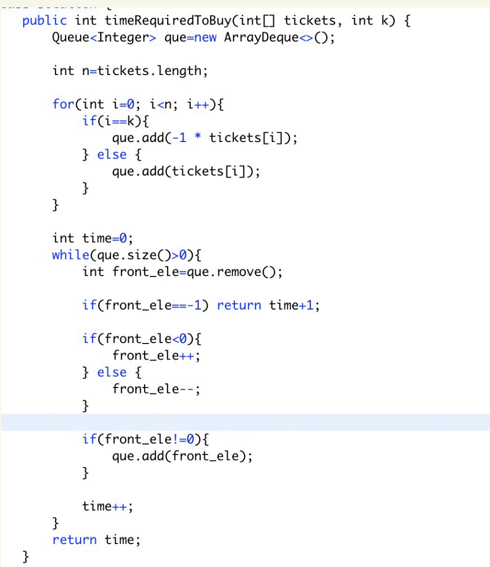
cs = ~~3~~ ~~2~~ ~~1~~ 0

, 1, 1, 1





```
    while(st.size()>0){
        int curr_students = que.size();

        while(curr_students>0 && que.peek()!=st.peek()){
            int front_ele=que.remove();
            que.add(front_ele);

            curr_students--;
        }

        if(curr_students==0) return st.size();

        st.pop();
        que.remove();
    }

    return 0;
```



```
class Solution {
    public int timeRequiredToBuy(int[] tickets, int k) {
        Queue<Integer> que=new ArrayDeque<>();

        int n=tickets.length;

        for(int i=0; i<n; i++){
            if(i==k){
                que.add(-1 * tickets[i]);
            } else {
                que.add(tickets[i]);
            }
        }

        int time=0;
        while(que.size()>0){
            int front_ele=que.remove();

            if(front_ele==-1) return time+1;

            if(front_ele<0){
                front_ele++;
            } else {
                front_ele--;
            }

            if(front_ele!=0){
                que.add(front_ele);
            }

            time++;
        }
        return time;
    }
```

k = 2

2, 3, 2

time = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

, 1

1

fc = -1

6

stack, queue using array