

# 알고리즘스터디 3주차 개인결과물 오윤기

\*회전하는 큐

초기상태 : 1 2 3 4 5 6 7 8 9 10

2번1회 : 2 3 4 5 6 7 8 9 10 1

1번연산 : 3 4 5 6 7 8 9 10 1

3번 3회 : 9 10 1 3 4 5 6 7 8

1번연산: 10 1 3 4 5 6 7 8

1. 2번 연산과 3번 연산의 횟수를 최소화 하기 위하여 시행 전에 왼쪽연산과 오른쪽 연산중 더 적은 횟수로 1번째 자리로 이동할 수 있는 연산의 종류를 판단한다

\* 판단방법: 현재 수행할 인덱스 i와 [(큐 사이즈) - i] 를 비교하여 i가 작은경우 2번연산, 그렇지 않은 경우 3번연산을 수행하도록 진행.

2. 1번의 과정을 진행했다면 1번연산을 수행.

3. 추출해야할 모든 원소가 남아 있지 않을 때 까지 1~3번 반복수행

4. 2번연산과 3번연산을 진행할 때마다 정적변수 CNT의 값을 올려주고 3번의 진행과정이 끝나면 CNT 출력

```
public class Main1021_RotateQueue {
    static int N,M;
    static int[] queue;
    static int size;
    static int totalCnt;
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        N = sc.nextInt();
        M = sc.nextInt();
        totalCnt = 0;
        queue = new int[N];
        for(int i = 0; i<N;i++)
            queue[i] = i+1;

        size = N;
        for(int i = 0; i<M;i++) {
            int tmp = sc.nextInt();
            for(int j = 0; j<size; j++) {
```

```

        if(tmp == queue[j]) {
            if(j == 0) {
                cal1(queue);
                break;
            }else {
                if(j<size-j) {
                    cal2(queue,j);
                    cal1(queue);
                }else {
                    cal3(queue,size-j);
                    cal1(queue);
                }
                break;
            }
        }
    }
}
System.out.println(totalCnt);
sc.close();
}

private static void cal1(int[] q) {
    for(int i = 0; i<size-1;i++)
        q[i] = q[i+1];
    size--;
}
private static void cal2(int[] q, int n) {
    for(int k = 0; k<n ;k++) {
        totalCnt++;
        int tmp = q[0];
        for(int i = 1; i <size; i++)
            q[i-1] = q[i];
        q[size-1] = tmp;
    }
}

private static void cal3(int[] q, int n) {
    for(int k = 0; k<n ;k++) {
        totalCnt++;
        int tmp = q[size-1];
        for(int i = size-2; i >=0; i--)
            q[i+1] = q[i];
        q[0] = tmp;
    }
}
}

```

\*풍선 터뜨리기

위치 1 2 3 4 5

종이값 3 2 1 -3 -1

3칸 왼쪽으로 이동하고 1있는 위치 지우기

위치 4 5 2 3

종이값 -3 -1 2 1

3칸 오른쪽으로 이동하고, 4위치 지우기

배열로 푼다면 출력할 위치를 첫번째 원소로 이동시키고

양수면 왼쪽 음수면 오른쪽으로 회전을 시킨다.

그리고 회전한 이후 출력할 번호의 원소를 지우고 반복한다.

들어오는 순서, 종이값을 멤버변수로 갖는 balloon 클래스 배열을 이용  
for(1 to N)

첫번째 원소에 있는 종이값만큼 회전

첫번째 위치 삭제(size--)