

# 1주차 - 1021: 회전하는 큐 풀이

202055524 김의현

2025-3-21

## 1 문제 설명

$a_1 \dots a_k$ 의 원소를 갖는 양방향 순환 큐가 있다. 이 큐는 다음과 같은 연산을 지원한다.

1. 첫번째 원소를 제거한다.  $a_1 \dots a_k$ 에서  $a_2 \dots a_k$ 가 된다.
2. 왼쪽으로 한 칸 이동한다.  $a_1 \dots a_k$ 에서  $a_2 \dots a_k, a_1$ 이 된다.
3. 오른쪽으로 한 칸 이동한다.  $a_1 \dots a_k$ 에서  $a_k, a_1 \dots a_{k-1}$ 이 된다.

큐中间的 원소를 제거하려면, 2번 혹은 3번 연산을 반복해 해당 원소를 큐의 첫번째에 위치시킨 후 제거하여야 한다.

크기  $N$ 인 큐에서 주어진  $M$ 개의 원소를 순서대로 제거하고자 한다. 이때 필요한 2번, 3번 연산의 최소값을 출력하는 프로그램을 작성하는 문제이다. (단,  $0 < M \leq N \leq 50$ )

## 2 접근법

입력이 작은 편이므로 시간초과를 걱정할 필요는 없다. 큐 앞/뒤에서 원소를 추가/삭제가 이뤄져야하므로 deque를 사용하기로 하였다.

2번 연산만 사용하는 경우 (원소의 위치)만큼 사용하면 되고, 3번 연산만 사용하는 경우 (큐의 크기) - (원소의 위치)만큼 사용하여야 한다. 어느 연산을 사용하는게 더 짧은지 판단한 다음, 이동하도록 구현하였다.

### 3 풀이

```
#include <cassert>
#include <deque>
#include <iostream>
#include <algorithm>
int main() {

    int queue_size, input_len;
    std::cin >> queue_size >> input_len;

    std::deque<int> deque;

    for (auto i = 1; i <= queue_size; ++i) {
        deque.push_back(i);
    }

    int cnt = 0;
    for (auto i = 0; i < input_len; ++i) {
        int input = 0;
        std::cin >> input;
        auto pos = std::find(deque.begin(), deque.end(), input) - deque.begin();
        auto rev_pos = deque.size() - pos;

        if (pos < rev_pos) { // move left
            cnt += pos;
            while (deque.front() != input) {
                auto tmp = deque.front();
                deque.pop_front();
                deque.push_back(tmp);
            }
        } else { // move right
            cnt += rev_pos;
            while (deque.back() != input) {

                auto tmp = deque.back();
```

```
        deque.pop_back();
        deque.push_front(tmp);

    }
    }
    assert(deque.front() == input);
    deque.pop_front();
}

std::cout << cnt;
}
```