

선형구조 알고리즘 문제해결 탐구_1

학번 : 1403

- 선정 자료구조 : (원형) 큐
- 난이도 : 하

문제 : 백준 1158번 요세푸스 문제

문제 작성
: 요세푸스 문제는 다음과 같다.
1번부터 N번까지 N명의 사람이 원을 이루면서 앉아있고, 양의 정수 K($\leq N$)가 주어진다. 이제 순서대로 K번째 사람을 제거한다. 한 사람이 제거되면 남은 사람들로 이루어진 원을 따라 이 과정을 계속해 나간다. 이 과정은 N명의 사람이 모두 제거될 때까지 계속된다. 원에서 사람들이 제거되는 순서를 (N, K)-요세푸스 순열이라고 한다. 예를 들어 (7, 3)-요세푸스 순열은 <3, 6, 2, 7, 5, 1, 4>이다.
N과 K가 주어지면 (N, K)-요세푸스 순열을 구하는 프로그램을 작성하시오.

입력	출력
첫째 줄에 N과 K가 빈 칸을 사이에 두고 순서대로 주어진다. ($1 \leq K \leq N \leq 5,000$)	예제와 같이 요세푸스 순열을 출력한다.

입력 예제	출력 예제
7 3	<3, 6, 2, 7, 5, 1, 4>
5 2	<2, 4, 1, 5, 3>
10 4	<4, 8, 2, 7, 1, 6, 10, 5, 9, 3>

- 문제 요약
 - 1번부터 N번까지 N명의 사람이 원을 이루며 앉아 있다. 양의 정수 K가 주어지면 순서대로 K번째 사람을 제거한다. 한 사람이 제거되면 남은 사람들은 다시 원을 이루며 앉고 제거된 사람 다음 사람부터 다시 K번째 사람을 찾아 제거한다. 모든 사람이 제거될 때까지 이 과정을 반복하고 제거되는 순서를 출력하는 문제이다.
- 조건 분석
 - N은 1 이상 5,000 이하이다.
 - K는 1 이상 N 이하이다.
 - 시간 제한 2초이다.
 - 메모리 제한 256MB이다.
 - 각 사람의 번호는 점표와 공백으로 구분되며 최종 결과는 < A, B, C ... > 의 형식으로 출력해야한다.
- 원형 큐 자료 구조를 선택한 근거

- 사람들이 순서대로 원을 그리며 앉아있고, 매번 K 번째 사람을 제거한다. 큐의 선입선출(FIFO) 특성을 활용하여 사람을 빼내고, 제거되지 않은 $K-1$ 명은 다시 큐의 맨 뒤로 보내며 K 번째 사람만 제거하는 방식이다.
- 사람들이 원을 이루고 있다는 문제의 조건은 순환적인 데이터 흐름을 의미한다. `CircularQueue`는 논리적으로 원형처럼 동작하여, 요소의 추가(enqueue)와 제거 (dequeue)를 빠르게 처리함으로써 문제의 원형 특성을 자연스럽게 반영하고 효율적인 연산을 할 수 있다.

● 의사 코드를 이용한 알고리즘 구현(주석을 이용하여 알고리즘을 구체적으로 설명)

```
from CircularQueue import CircularQueue

N, K = map(int, input().split()) # N: 사람의 수, K: 제거할 순서

queue = CircularQueue(N) # N명의 사람을 담을 수 있는 CircularQueue 생성 (실제 용량 N+1)

# 1. 1번부터 N번까지의 사람을 큐에 순서대로 넣음.
for i in range(1, N + 1):
    queue.enqueue(i)

result = [] # 제거된 사람들을 저장할 리스트

# 2. 큐가 빌 때까지 다음 과정을 반복.
while not queue.isEmpty():
    # 2-1. K-1번째까지의 사람들은 제거하지 않고 큐의 맨 뒤로 다시 보냄.
    for _ in range(K - 1):
        person = queue.dequeue() # 맨 앞에 있는 사람을 꺼내서
        queue.enqueue(person) # 다시 맨 뒤로 넣음.

    # 2-2. K번째 사람은 큐에서 제거(dequeue)하여 결과 리스트에 추가.
    removed_person = queue.dequeue()
    result.append(removed_person)

# 3. 모든 사람이 제거된 후, 결과 리스트를 요구하는 형식으로 출력.
print("<" + ", ".join(map(str, result)) + ">")
```

● 시간복잡도 : $O(N * K)$

- 총 N 명의 사람을 모두 제거해야 하며 각 사람을 제거하기 위해 최대 K 번의 요소 이동(건너뛰기)이 반복된다. 큐의 모든 연산 (enqueue, dequeue)은 $O(1)$ 의 시간이 걸리므로, 총 연산량은 $N * K$ 에 비례하여 $O(N * K)$ 의 시간 복잡도를 가진다.