

# “레벨 업” 문제 풀이

작성자: 이민제

## 부분문제 1

캐릭터를 레벨이 낮은 순서대로 정렬하고, 가장 레벨이 낮은  $K$ 명의 캐릭터의 레벨을 올리는 방식으로 레벨 업 과정을 직접 시뮬레이션할 수 있다. 시간 복잡도는  $O(MN \log N)$ 이다.

## 부분문제 2

캐릭터를 레벨이 낮은 순서대로 정렬하자.  $K = 1$ 인 경우에는 트레이닝이 다음과 같이 이루어짐을 알 수 있다.

- 첫 번째 캐릭터의 레벨이 계속해서 오른다.
- 첫 번째와 두 번째 캐릭터의 레벨이 같아지면, 이후 이 두 캐릭터의 레벨이 번갈아 가며 오른다.
- 첫 번째, 두 번째, 세 번째 캐릭터의 레벨이 같아지면, 이후 이 세 캐릭터의 레벨이 번갈아 가며 오른다.
- 이러한 과정은 모든 캐릭터의 레벨이 같아질 때까지 계속되며, 이후에는  $N$ 개의 캐릭터의 레벨이 번갈아 가며 오른다.

각  $1 \leq i \leq N$ 에 대해, 첫  $i$ 개 캐릭터의 레벨이 같아지는 시점을 누적 합 등을 이용하여 계산할 수 있다. 이 시점이  $M$ 을 초과하지 않으면서 최대가 되도록 하는  $i$ 를  $k$ 라 하자. 이때  $(k+1)$ 번째,  $\dots$ ,  $N$ 번째 캐릭터는 레벨이 오르지 않음을 알 수 있다. 한편 1번째 캐릭터와  $k$ 번째 캐릭터의 레벨 차이는 최대 1이므로, 전체 캐릭터의 레벨 총합을 이용하여 각각의 레벨을 구할 수 있다.

시간 복잡도는  $O(N \log N)$ 이다.

## 부분문제 3

부분문제 3에서는  $M$ 이 크지 않으므로, 각각의 트레이닝을 빠르게 시뮬레이션하는 방식으로 문제를 해결할 수 있다.

캐릭터가 레벨이 낮은 순서대로 정렬되어 있다고 가정하고,  $K$ 번째 캐릭터의 레벨을  $x$ 라고 하자. 또, 레벨이  $x$  미만인 캐릭터의 수를  $l$ ,  $x$  이하인 캐릭터의 수를  $r$ 라고 하자. 트레이닝을 한 번 진행하면 캐릭터의 레벨은 다음과 같이 변한다.

- 1번째,  $\dots$ ,  $l$ 번째 캐릭터의 레벨이 1 오른다.
- $(l+1)$ 번째,  $\dots$ ,  $r$ 번째 캐릭터 중  $(K-l)$ 명의 캐릭터의 레벨이 1 오른다. 편의상  $(l+r-K+1)$ 번째,  $\dots$ ,  $r$ 번째 캐릭터의 레벨이 오른다고 하자. 그러면 트레이닝 이후에도 캐릭터들의 레벨이 정렬된 상태를 유지하게 된다.

따라서 정렬된 배열에서 다음과 같은 작업을 효율적으로 수행할 수 있으면 된다.

- 위에서 정의한  $x$ ,  $l$ ,  $r$ 을 구한다.

- 구간  $[1, l]$ ,  $[(l + r - K + 1), r]$ 에 1을 더한다.

이는 Lazy Propagation을 사용한 Segment Tree로 빠르게 처리할 수 있다. 시간 복잡도는  $O(M \log N)$ 이다.

## 부분문제 4

부분문제 2와 마찬가지로, 트레이닝이 진행됨에 따라 캐릭터들의 레벨이 점점 일정해진다는 사실을 관찰할 수 있다. 단, 부분문제 2에서는 레벨이 가장 낮은 캐릭터를 중심으로 레벨이 일정해졌지만, 일반적인 경우에는 레벨이  $K$ 번째로 낮은 캐릭터를 중심으로 레벨이 일정해진다.

맨 처음 상태에서  $K$ 번째로 낮은 레벨을  $x$ 라 할 때,  $N$ 명의 캐릭터를 다음과 같이 세 그룹으로 나누자.

- A그룹: 레벨이  $x$  미만.
- B그룹: 레벨이  $x$  이상  $x + 1$  이하.
- C그룹: 레벨이  $x + 1$  초과.

트레이닝을 진행하면 각 그룹에서 캐릭터들의 레벨은 다음과 같이 변한다.

- A그룹: 모든 캐릭터의 레벨이 1 오른다.
- B그룹: 레벨이 낮은  $(K - |A|)$ 명의 레벨이 1 오른다. (단,  $|A|$ 는 그룹 A의 크기.)
- C그룹: 레벨이 오르지 않는다.

트레이닝을 진행하다 보면 A그룹 혹은 C그룹의 캐릭터가 B그룹의 레벨 범위 안으로 들어오는 경우가 발생하는데, 이 경우 해당 캐릭터를 B그룹으로 옮기자. 이때 다음의 사실들을 관찰할 수 있다.

- 레벨이  $K$ 번째로 낮은 캐릭터는 항상 B그룹에 속한다. 즉, 위에서 서술한 각 그룹의 성질은 계속해서 유지된다.
- B그룹 내에서 캐릭터 간의 레벨 차이는 1 이하이다. 따라서 B그룹의 크기와 레벨 범위, 각 레벨에 해당하는 캐릭터의 수를 알면 B그룹 전체의 정보를 얻게 된다. 이 값들을 관리하자.
- A그룹의 캐릭터는 레벨이 높은 순서대로 B그룹으로 들어오며, C그룹의 캐릭터는 레벨이 낮은 순서대로 B그룹으로 들어온다.

A그룹에서 레벨이 가장 높은 캐릭터 또는 C그룹에서 레벨이 가장 낮은 캐릭터가 B그룹으로 들어오는 과정을 시뮬레이션하자. 이 캐릭터가 B그룹으로 들어오는 시점은 앞서 관리한 B그룹의 정보를 바탕으로 이분 탐색(Parametric Search)을 이용하여 계산할 수 있다.

이를 반복하여 각 캐릭터가 B그룹으로 들어오는 시점을 모두 구하자. B 그룹으로 새로운 캐릭터가 들어오는 경우가 총  $O(N)$ 번 발생하므로, 위 과정을  $O(N \log M)$  시간에 모두 시뮬레이션할 수 있다.

트레이닝이 종료된 시점에서 각 캐릭터가 어느 그룹에 속하는지 안다면, 각각의 레벨 또한 어렵지 않게 구할 수 있다. 총 시간복잡도는  $O(N(\log N + \log M))$ 이다.