

BOJ 8217

유성

#parallel_binary_search

난이도 – **Diamond IV**

BOJ 8217 유성

- 구간 업데이트 / 모든 점에 대한 점 쿼리이므로 imos 펜윅 트리를 사용합니다.
- 각각의 점 쿼리는 시간에 대한 단조성을 가집니다. 따라서 일 수를 기준으로 이분 탐색을 수행하면 빠르게 답을 찾을 수 있습니다.
- 이 경우 시간 복잡도는 대략 $O(NQ \lg M \lg Q)$ 가 됩니다. TLE입니다.

BOJ 8217 유성

- 시간을 줄여야 하는데, 펜윅 트리로 매번 시뮬레이션을 돌려 이분 탐색을 하는 방법은 반드시 $O(Q \lg Q \lg M)$ 이상의 시간이 소모됩니다.
- 이보다 더 빠르게 각 쿼리를 처리하는 방법은 찾기 어렵습니다.
- 그러니까, 시간 복잡도에서 **N을 떼 버리는 방법**을 생각해 봅시다.

BOJ 8217 유성

- 시간이 많이 걸리는 원인은 매번 펜윅 트리를 새로 구축해야 하기 때문입니다.
그렇다면 한 번 만든 펜윅 트리를 재활용하면 어떨까요?
- 처음에 mid 일까지의 업데이트를 처리한 펜윅 트리는
N개의 국가 모두 시뮬레이션 해 보아야 하므로 이 트리를 새로 만들지 말고
N개의 국가 모두에 대해 한 번에 확인해 주는 방법을 이용합시다.
- 요컨대 N개의 점에 대한 답을 **동시에** 구하는 것입니다.
- 이러한 기법을 **병렬 이분 탐색**이라 부릅니다.

BOJ 8217 유성

- N개의 국가에 대한 left와 right를 모두 만들어주고, 각각에 대한 이분 탐색을 동시에 진행합니다.
- 만약 left와 right가 모두 같은 구간이 여러 개라면, 해당 경우에 대해서는 트리를 한 번만 구성해 주어도 됩니다.
- 모든 점에 대한 답을 얻을 때까지 이를 반복합니다.
- 총 시간 복잡도는 $O((Q+M) \lg M \lg Q)$ 가 됩니다.