

BOJ 10070



#segtree #lazy\_prop

난이도 – **Diamond V**

## BOJ 10070 벽

- 주어지는 쿼리는 다음과 같습니다.
  - 구간  $[L, R]$ 의 각 원소  $A_L, \dots, A_R$ 에 대해  $A_i$ 를  $\max(A_i, v)$ 로 변경
  - 구간  $[L, R]$ 의 각 원소  $A_L, \dots, A_R$ 에 대해  $A_i$ 를  $\min(A_i, v)$ 로 변경
- 구간 업데이트 점 쿼리지만  $\max, \min$  연산은 역원이 없기 때문에 펜윅 트리는 이용할 수 없습니다.
- 느리게 갱신되는 세그먼트 트리를 이용합시다.

## BOJ 10070 벽

- 구간의 최솟값과 최댓값을 저장하는 세그먼트 트리를 각각 만듭니다.
- 각 노드에 저장된 값은 모두 0으로 초기화되어 있으면 됩니다.
- lazy 배열은 따로 잡아주지 않아도 좋습니다.

(이 문제에선 세그먼트 트리 자체를 lazy 배열처럼 사용할 수 있습니다.)

## BOJ 10070 벽

전파는 다음과 같이 이루어집니다.

- 최댓값 쿼리인 경우, 두 세그먼트 트리의 노드에 각각 최댓값 연산을 합니다.
- 최솟값 쿼리인 경우, 두 세그먼트 트리의 노드에 각각 최솟값 연산을 합니다.

## BOJ 10070 벽

**업데이트**는 다음과 같이 이루어집니다.

- 쿼리로 주어진 구간을  $[L, R]$ , 현재 업데이트하는 구간을  $[S, E]$ , 현재 노드의 인덱스를  $NOW$ , 쿼리로 주어진 값을  $V$ 라고 합시다.
- 보통의 세그먼트 트리와 같이, 두 구간이 만나지 않으면 return합니다.
- $[L, R]$ 이  $[S, E]$ 를 포함하는 경우,  $NOW$ 에  $V$ 를 전파해 줍니다.
  - 이때 정답 배열  $A$ 를 만들어서,  $A[S]$ 에 **최댓값 세그먼트 트리의  $NOW$ 에 저장된 값을 저장**해 줍니다.

## BOJ 10070 벽

- 이외의 경우, 구간  $[L, R]$ 을  $[L, \text{mid}]$ 와  $[\text{mid}+1, R]$ 로 나누어 줍시다.
- 나누어 준 각각의 구간에 대한 노드에 다음과 같이 전파를 수행합니다.
  - 각 구간의 최댓값 세그먼트 트리 노드에  
NOW에 저장된 값에 대한 **최댓값 쿼리를 전파**
  - 각 구간의 최솟값 세그먼트 트리 노드에  
NOW에 저장된 값에 대한 **최솟값 쿼리를 전파**

## BOJ 10070 벽

- 전파 후 다른 쿼리가 들어왔을 때의 처리를 위해  
현재 구간의 최댓값 세그먼트 트리 노드 값을 적당히 작은 값(-1 등)으로,  
최솟값 세그먼트 트리 노드 값을 적당히 큰 값( $1e9$  등)으로 초기화해 줍니다.
- 이제 나누었던 각 구간에 대한 업데이트를 수행해 주면 됩니다.
- 각 업데이트마다 시간 복잡도는  $O(\lg N)$ 입니다.

## BOJ 10070 벽

- K개의 쿼리를 모두 처리한 후, A 배열을 출력해야 합니다.
- 그 전에, 아직 전파되지 않은 값들을 모두 전파해 주어야 합니다.
- 0부터 N-1까지의 모든 구간  $[i, i]$ 에 대해 최댓값 쿼리를 넣어 줍니다.
  - 쿼리로 주어지는 값은 적당히 작은 수로 해 주어야  
결과에 영향을 주지 않습니다.
- 이제 업데이트에 의해  $A[i]$ 에 모든 쿼리가 전파된 후의 값이 저장됩니다.
- 총 시간 복잡도는  $O((N+K) \lg N)$  입니다.