

BOJ 18798 OR과 쿼리

#segtree

난이도 – Diamond V

BOJ 18798 OR과 쿼리

- 구간 OR 업데이트 / 구간 쿼리를 수행해야 합니다.
- 다양한 풀이 방법이 존재합니다.

BOJ 18798 OR과 쿼리

#1. Segment Tree with Lazy Propagation

- 구간 업데이트 / 구간 쿼리를 처리하면 되므로
레이지 세그먼트 트리를 잘 활용하면 해결할 수 있습니다.
- 그러나, 느린 전파 **없이** 문제를 해결할 수 있는 방법이 존재합니다.

BOJ 18798 OR과 쿼리

#2. Segment Tree without Lazy Propagation

- 2번 쿼리는 그냥 구간 안의 K의 개수를 세는 세그를 통해 처리할 수 있습니다.
- 레이지 없이 1번 쿼리를 어떻게 빠르게 처리할 수 있을까요?

BOJ 18798 OR과 쿼리

- 레이지를 이용하지 않고 문제를 해결하려면 중요한 관찰 하나가 필요한데, OR 연산이 **절대로 수를 감소시키지 않는다**는 점입니다.
- 왜냐하면, 한 번 1이 된 비트가 OR 연산을 통해 0이 될 수 없습니다.
이 성질을 활용합니다.

BOJ 18798 OR과 쿼리

- 구간 안의 모든 수를 AND한 값을 저장하는 세그먼트 트리를 만듭니다.
- 이제 OR 업데이트를 처리할 때, 이 트리의 노드의 값과 쿼리로 주어진 값을 OR했을 때 노드의 값이 변하지 않았다면, 해당 업데이트를 넘어갑니다.
- 결과적으로 각 원소마다 업데이트는 최대 30번까지만 일어나게 됩니다.
- 따라서 1번 쿼리는 모두 합쳐 최대 $O(N)$ 에 처리됩니다.
- 총 시간 복잡도는 $O(N + Q \lg N)$ 입니다.

BOJ 18798 OR과 쿼리

- 이외에 제공된 분할법 등의 최적화 기법을 통해 해결할 수도 있습니다.