

“고기 파티” 문제 풀이

작성자: 김동현

부분문제 1

각 사람마다, 지금 불판에 남아 있는 각 고기에 대해 몇 개의 꼬치가 이 고기를 찌르는지 일일이 세고, 2개의 꼬치에 찔린 고기들의 맛의 합을 구한 뒤 1개 이상의 꼬치에 찔린 모든 고기를 제거해주면 된다. $O(NM)$ 시간에 문제를 해결할 수 있다.

부분문제 2

같은 (s_i, e_i) 를 가지는 고기를 하나로 묶어서 생각하면, $e_i - s_i \leq 5$ 이므로 각 꼬치마다 해당 꼬치가 찌를 수 있는 고기는 최대 15종류임을 알 수 있다. `std::map` 등의 자료 구조를 이용하여 각 꼬치마다 해당하는 고기들 중 아직 찔리지 않은 것을 모두 찾아서 뽑아낸 뒤 두 꼬치에 모두 찔리는 고기는 답에 반영해주면 된다.

부분문제 3

i 번째 고기의 구간은 $i + 1$ 번째 고기의 구간을 포함하므로, $i + 1$ 번째 고기가 어느 시점에 꼬치에 찔렸다면 i 번째 이전의 모든 고기도 같이 찔리게 된다. 즉, 임의의 순간에 고기가 1개 이상 남아있다면 어떤 k ($1 \leq k \leq N$)에 대해 $k, k + 1, \dots, N$ 번째 고기들이 남아있는 형태일 것이다.

따라서, 고기들을 입력 받은 순서대로 큐(Queue) 자료구조에 넣은 뒤, 각 사람이 들어올 때마다 큐의 맨 앞에 있는 고기가 1개 이상의 꼬치에 찔린다면 pop을 수행해주는 것을 불가능할 때까지 반복하면 된다. 한 번 빠진 고기는 더 이상 고려되지 않으므로 총 시간복잡도는 $O(N + M)$ 이다.

부분문제 4

$e_i - s_i = L$ 이라 하면, 각 고기가 차지하는 구간을 $[s_i, s_i + L]$ 로 나타낼 수 있다. 따라서, 각 고기의 구간에 대한 정보를 s_i 만 가지고도 나타낼 수 있다.

$a_j + 0.1$ 좌표에 찌른 꼬치는 $a_j - L + 1 \leq s_i \leq a_j$ 를 만족하는 고기를 찌르고, $b_j + 0.9$ 좌표에 찌른 꼬치는 $b_j - L + 1 \leq s_i \leq b_j$ 를 만족하는 고기를 찌른다. 따라서, 해당 범위에 속하는 고기들을 빠르게 찾아내고 제거할 수 있다면 문제를 효율적으로 해결할 수 있다.

`std::set` 등의 균형 이진 트리 기반의 자료 구조를 사용하자. $a_j - L + 1 \leq s_i$ 이면서 최소인 s_i 를 자료 구조에서 찾고 제거하는 것을 $O(\log N)$ 에 할 수 있으므로, 해당하는 고기가 없거나 $s_i > a_j$ 가 되기 전까지 반복하면 $O(\text{범위 내에 남아있는 고기 수} + 1) \times \log N$ 시간복잡도에 특정 꼬치에 찔린 모든 고기를 자료구조에서 뽑아낼 수 있다. 두 꼬치 각각에 대해 이를 수행한 뒤, 뽑아낸 고기들 중 두 꼬치에 모두 찔린 고기들을 답에 더해지면 된다.

한 번 자료구조에서 뽑은 고기는 다시 고려하지 않으므로, 총 시간복잡도는 $O((N + M) \log N)$ 이다.

부분문제 5

부분문제 4의 아이디어를 약간 확장하여, 일반적인 경우에 대해서도 특정 꼬치에 찔리는 고기를 $O(\text{찔린 고기 수} \times \log N)$ 정도의 시간에 모두 뽑아낼 수 있다. 매우 다양한 방법으로 해결이 가능하며, 이 중 두 가지를 소개한다.

첫 번째 방법은 구간의 최댓값과 그 위치를 같이 관리하는 세그먼트 트리를 이용한다. 우선 주어진 고기들을 s_i 가 증가하는 순서대로 정렬하자. 그리고 세그먼트 트리의 인덱스 i 에 해당하는 값을 e_i 로 초기화한다. 어떤 꼬치를 좌표 $X + 0.1$ 에 찌른다고 하자. 이 때 이 꼬치에 찔리는 고기는 $s_i \leq X$, $X + 1 \leq e_i$ 를 만족하는 고기들이다. ($X + 0.9$ 에 찌르는 경우에도 동일하다.)

u 를 $s_u \leq X$ 를 만족하는 최대 인덱스라 하자. 세그먼트 트리의 $[1, u]$ 구간에서 최댓값을 찾고, 그 값이 $X + 1$ 이상이라면 뽑아내는 것을 반복하자. 뽑아낼 때는 해당하는 고기의 인덱스가 k 일 때, 인덱스 k 에 해당하는 값을 -1 로 설정하면 된다. 이것을 최댓값이 $X + 1$ 미만이 될 때까지 반복하게 되면 $O((\text{찔린 고기 수} + 1) \times \log N)$ 시간만에 특정 꼬치가 찌르는 모든 고기를 뽑을 수 있다. 총 시간복잡도는 부분문제 4와 마찬가지로 $O((N + M) \log N)$ 이다.

두 번째 방법은 세그먼트 트리의 각 노드마다 리스트를 관리하는 것이다. 각 고기가 차지하는 구간들의 양 끝점을 가지고 좌표 압축을 수행하면, 각 고기가 압축된 좌표에서 차지하는 구간을 세그먼트 트리 상에서 $O(\log N)$ 개의 노드로 쪼갤 수 있으므로 해당하는 노드들의 리스트에 각각 고기 번호를 삽입한다. 이렇게 하면 고기 번호를 자료구조에 총 $O(N \log N)$ 개만큼 넣게 된다.

이제 어떤 꼬치가 찌르는 고기를 모두 뽑아내기 위해서는, 해당 꼬치의 (압축된) 좌표를 포함하는 $O(\log N)$ 개의 노드들의 리스트에 들어 있는 고기 번호들을 모두 뽑아낸 뒤, 이전에 찔린 적이 없는 고기들만 취하면 된다. 한 번 번호를 뽑아낸 리스트는 빈 리스트가 되므로 뽑아내는 고기 번호는 많아야 $O(N \log N)$ 개이다. 고기 번호를 뽑아내는 것은 리스트 순회이고, 이전에 찔린 적이 없는지 체크하는 것은 boolean 배열로 가능하므로 고기 번호 하나당 $O(1)$ 만에 모든 필요한 처리가 가능하다. 총 시간복잡도는 $O((N + M) \log N)$ 이다.