중등부 3. 버블버블

풀이 작성자: 윤창기

$X = \max(A_1, \cdots, A_N)$이라고 하자. 모든 풀이에 앞서, 수 하나를 바꿀 때는 1 이상 X 이하의 정수값에 대해서 시도해보면 충분하다는 사실을 알 수 있다. 정수가 아닌 실수로 수를 바꾸는 경우, 가장 가까운 정수로 답을 바꾸어도 교환 횟수가 변하지 않는다.

부분문제 1 ($N \le 200$, $X \le 200$)

수 하나를 바꿀 수 있는 경우가 $O(NX)$가지이므로, 모든 경우에 대해 버블 정렬을 직접 시도하는 경우 $O(N^3X)$의 시간복잡도로 구현에 따라 점수를 받을 수 있다. 추가적으로 버블 정렬의 교환 횟수는 "$i < j$이지만 $A_i > A_j$인 $1 \le i < j \le N$의 개수"와 같으며, 이를 수열의 inversion count라고 한다. 단순히 inversion count를 계산하는 방법은 $O(N^2)$시간이 걸리지만, A_i를 변경할 때는 i번째 수와 관련된 inversion에만 변화가 생긴다는 점을 관찰하여 $O(N^2X)$에 문제를 해결할 수도 있다.

부분문제 2 ($N \le 3\,000$, $X \le 3\,000$)

수열의 inversion count는 다음 질의를 처리할 줄 알면 쉽게 구할 수 있다.

- 현재 집합에서 x보다 작은 수의 개수를 반환한다. ($1 \le x \le X$)
- 집합에 x를 하나 넣는다. ($1 \le x \le X$)

두 질의는 세그먼트 트리, 펜윅 트리 등의 자료구조를 이용하여 매번 $O(\log X)$의 시간에 답할 수 있으며, 때문에 $O(N \log X)$ 시간에 수열의 inversion count를 구할 수 있다. 이를 이용하여 $A_i = x$일 때 inversion count가 어떻게 바뀌는지를 위 질의를 응용하여 구할 수 있으며, 가능한 모든 경우에 대해 시도함으로써 $O(NX \log X)$의 시간복잡도에 문제를 해결할 수 있다.

부분문제 3 ($N \le 15\,000$, $X \le 15\,000$)

($1 \le i \le N, 1 \le j \le X$)에 대해 다음의 두 수열을 정의하자.

- $G_{i,j}$: $A_1, \cdots, A_{i-1}$중 j보다 큰 수의 개수
- $L_{i,j}$: $A_{i+1}, \cdots, A_n$ 중 j보다 작은 수의 개수

정의로부터 수열의 inversion count는 $I = \sum_{i=1}^{n} G_{i,A_i}$가 됨을 알 수 있다. 또한, A_i를 j로 바꿨을 때 inversion count는 $I - (G_{i,A_i} + L_{i,A_i}) + (G_{i,j} + L_{i,j})$가 됨을 알 수 있다. 따라서 G, L을 이용하여 가능한 $O(NX)$가지의 경우마다 $O(1)$ 시간에 inversion count를 계산할 수 있다. 또한, G, L의 경우 간단한 dynamic programming 방법으로 총 $O(NX)$ 시간에 계산할 수 있다. G, L을 모두 저장하는 데 많은 메모리가 필요할 수 있으나, 이는 $G_{i,j}$를 구할 때 $G_{i-1,j}$의 값만 필요하다는 사실을 이용하면 필요한 메모리를 $O(N)$으로 줄일 수 있다.

부분문제 4 ($N \le 300\,000$, $X \le 1\,000\,000$)

바꿀 위치 j를 고정하고, $i < j$에 대해 A_i가 inversion count를 얼마나 늘리는지 생각해보자. A_j를 $[1, A_i - 1]$의 수 중 하나로 바꾸는 경우 1, $[A_i, X]$의 수 중 하나로 바꾸는 경우 0만큼 inversion count가 늘어난다. 마찬가지로 $k > j$에 대해서는 A_j가 $[1, A_k]$에 들어가도록 바꿀 때 0, $[A_k + 1, A_n]$에 들어가도록 바꿀 때 1만큼 inversion count가 늘어난다. 따라서 lazy propagation을 지원하는 최솟값 세그먼트 트리를 이용하여, A_j를 변경했을 때 inversion count의 최솟값을 $O(N \log X)$ 또는 $O(N \log N)$ 시간에 구할 수 있다.