수열 정렬하기 – 풀이

작성자: 장근영

기본 관찰

한 번의 시행에서 기준값 x를 고르면, 수열은 $[A_i \le x]$와 $[A_i > x]$를 이어 붙인 형태가 된다. 여기서 각각의 부분수열 안에서는 원래 수열에서의 상대적인 순서가 유지된다.

여러 번의 시행에서 선택한 기준값들을 생각하자.

먼저 같은 기준값을 여러 번 선택하는 것은 의미가 없다. 한 번 x를 기준으로 나누고 나면, 이미 모든 x 이하의 원소가 모든 x 초과의 원소보다 앞에 있으므로, 다시 같은 x를 선택해도 수열은 변하지 않는다.

또한 서로 다른 두 기준값 x, y에 대해, 이 둘을 적용하는 순서도 중요하지 않다. $x < y$라고 하자.

먼저 x를 적용한 뒤 y를 적용하면 다음과 같다. x를 적용한 뒤 수열은 $[A_i \le x] + [A_i > x]$가 된다. 이제 y를 적용하면, 앞쪽의 $[A_i \le x]$에 속한 원소들은 모두 y 이하이므로 그대로 앞에 남는다. 뒤쪽의 $[A_i > x]$ 중에서는 $x < A_i \le y$인 원소들이 그다음으로 오고, $y < A_i$인 원소들이 마지막에 온다. 따라서 최종 수열은 $[A_i \le x] + [x < A_i \le y] + [y < A_i]$가 된다.

반대로 먼저 y를 적용한 뒤 x를 적용해도 같은 결과가 나온다. y를 적용한 뒤 수열은 $[A_i \le y] + [A_i > y]$가 된다. 이제 x를 적용하면, 앞쪽의 $[A_i \le y]$ 안에서 $A_i \le x$인 원소들이 먼저 오고, 그다음에 $x < A_i \le y$인 원소들이 온다. 뒤쪽의 $[A_i > y]$에 속한 원소들은 모두 x 초과이므로 마지막에 남는다. 따라서 이 경우에도 최종 수열은 $[A_i \le x] + [x < A_i \le y] + [y < A_i]$가 된다.

각 단계에서 같은 범위에 속한 원소들끼리는 원래 수열에서의 상대적인 순서가 계속 유지된다. 따라서 두 기준값 x, y를 적용한 결과는 적용 순서와 무관하다.

결국 여러 번의 시행에서 선택한 기준값들은 순서를 바꾸어도 최종 결과가 변하지 않는다. 따라서 선택한 기준값들을 중복 없이 정렬해서 $x_1 < x_2 < \cdots < x_k$라고 두어도 된다.

이제 이 기준값들을 오름차순으로 적용한다고 생각하자. 처음 x_1을 적용하면 수열은 $[A_i \le x_1] + [x_1 < A_i]$가 된다. 그다음 x_2를 적용하면 첫 번째 부분 $[A_i \le x_1]$은 모두 x_2 이하이므로 그대로 앞에 남고, 뒤쪽 부분 $[x_1 < A_i]$이 $[x_1 < A_i \le x_2]$와 $[x_2 < A_i]$로 나뉜다.

이 과정을 반복하면, 최종 수열은 $[A_i \le x_1]$, $[x_1 < A_i \le x_2]$, $\ldots$, $[x_{k-1} < A_i \le x_k]$, $[x_k < A_i]$를 차례대로 이어 붙인 수열이 된다.

즉, 시행들의 순서는 중요하지 않고, 선택한 기준값들의 집합만 중요하다. 따라서 문제는 값의 범위 $[1, N]$을 몇 개의 연속한 구간으로 나누어서, 각 구간에 속한 값들만 원래 순서대로 뽑은 부분수열이 비내림차순이 되게 만드는 문제로 볼 수 있다.

값의 구간 수가 c개라면, 그 구간들을 만들기 위해 필요한 시행 횟수는 $c - 1$번이다.

부분문제 1

모든 원소가 1 또는 2이다.

이미 모든 1이 모든 2보다 앞에 있다면 수열은 비내림차순이므로 답은 0이다.

그렇지 않다면 어떤 2가 어떤 1보다 앞에 있다. 이 경우 $x = 1$로 한 번 시행하면 모든 1이 앞으로 가고, 모든 2가 뒤로 가므로 수열이 정렬된다.

따라서 수열에 2가 1보다 앞에 등장하는 경우가 있으면 답은 1, 그렇지 않으면 답은 0이다.

전체 시간 복잡도는 $\mathcal{O}(N)$이다.

부분문제 2

$N \le 15$이다.

의미 있는 기준값은 $1, 2, \ldots, N-1$뿐이다. 따라서 가능한 기준값의 부분집합을 모두 시도할 수 있다.

각 부분집합 S에 대해, S의 원소들을 오름차순으로 적용한 결과를 직접 만들어 본다. 그 결과가 비내림차순이면, $|S|$가 가능한 시행 횟수이다.

모든 부분집합을 확인하면서 가능한 $|S|$의 최솟값을 구하면 된다.

부분집합의 개수는 최대 2^{N-1}개이고, 각 경우에 대해 시행을 직접 시뮬레이션하면 충분하다.

전체 시간 복잡도는 $\mathcal{O}(2^N N^2)$이다.

부분문제 3

$N \le 100$이다.

값의 구간 $[l, r]$에 대해, 값이 l 이상 r 이하인 원소들만 원래 순서대로 뽑았을 때 그 부분수열이 비내림차순인지 확인하자. 이를 $\mathrm{ok}(l, r)$라고 하자.

$\mathrm{ok}(l, r)$를 가장 단순하게 계산하려면, 모든 위치 쌍 $p < q$를 확인하면 된다.

만약 $l \le A_p \le r$, $l \le A_q \le r$이면서 $A_p > A_q$인 쌍이 존재한다면, 구간 $[l, r]$만으로 이루어진 부분수열은 비내림차순이 될 수 없다. 그런 쌍이 없다면 가능하다.

하나의 (l, r)에 대해 모든 위치 쌍을 확인하는 데 $\mathcal{O}(N^2)$ 시간이 걸린다. 가능한 (l, r) 쌍은 $\mathcal{O}(N^2)$개이므로, 모든 $\mathrm{ok}(l, r)$를 계산하는 데 $\mathcal{O}(N^4)$ 시간이 걸린다.

이제 DP를 하자. $\mathrm{dp}[i]$를 값 $1, 2, \ldots, i$를 몇 개의 연속한 값 구간으로 나누어야 하는지의 최솟값이라고 하자.

초기값은 $\mathrm{dp}[0] = 0$이다.

마지막 구간이 $[j+1, i]$라고 하면, 이 구간은 $\mathrm{ok}(j+1, i)$가 참이어야 한다. 따라서 $0 \le j < i$인 모든 j 중에서 가능한 것을 골라 $\mathrm{dp}[i] = \min(\mathrm{dp}[j] + 1)$로 계산할 수 있다.

최소 시행 횟수는 구간 수보다 1 작으므로, 정답은 $\mathrm{dp}[N] - 1$이다.

전체 시간 복잡도는 $\mathcal{O}(N^4)$이다.

부분문제 4

$N \le 750$이다.

부분문제 3과 같은 DP를 사용하되, $\mathrm{ok}(l, r)$를 더 빠르게 계산한다.

값의 구간 $[l, r]$에 대해, 수열 A를 왼쪽에서 오른쪽으로 한 번 훑는다. 이때 $l \le A_i \le r$인 원소만 보면서, 직전에 선택된 값보다 현재 값이 작아지는 순간이 있는지 확인한다.

현재까지 선택한 마지막 값을 `last`라고 하자. $l \le A_i \le r$인 원소를 만났을 때 `last` $> A_i$라면, $\mathrm{ok}(l, r)$는 거짓이다. 그렇지 않다면 `last`를 A_i로 갱신한다.

이렇게 하면 하나의 (l, r)에 대해 $\mathrm{ok}(l, r)$를 $\mathcal{O}(N)$ 시간에 계산할 수 있다.

가능한 (l, r) 쌍은 $\mathcal{O}(N^2)$개이므로, 모든 $\mathrm{ok}(l, r)$를 계산하는 데 $\mathcal{O}(N^3)$ 시간이 걸린다.

이후 DP는 부분문제 3과 동일하다. $\mathrm{dp}[0] = 0$으로 두고, 마지막 구간 $[j + 1, i]$가 가능한 모든 j에 대해 $\mathrm{dp}[i] = \min(\mathrm{dp}[j] + 1)$을 계산한다.

정답은 $\mathrm{dp}[N] - 1$이다.

전체 시간 복잡도는 $\mathcal{O}(N^3)$이다.

부분문제 5

모든 원소가 서로 다르다.

수열의 길이가 N이고, 모든 값이 1 이상 N 이하이므로, 수열 A는 $1, 2, \ldots, N$의 순열이다.

값 i가 등장하는 위치를 p_i라고 하자.

값 i와 $i + 1$이 같은 값 구간에 들어간다고 하자. 같은 구간 안에서는 원래 수열에서의 상대적인 순서가 유지된다.

따라서 $p_i > p_{i+1}$이면, 원래 수열에서 $i + 1$이 i보다 앞에 있으므로 i와 $i + 1$을 같은 구간에 둘 수 없다. 즉, 반드시 i와 $i + 1$ 사이를 나누는 기준값을 골라야 한다.

반대로 $p_i < p_{i+1}$이면, i와 $i + 1$ 사이를 나누지 않아도 이 둘의 순서는 올바르다.

따라서 답은 $p_i > p_{i+1}$인 i의 개수이다.

전체 시간 복잡도는 $\mathcal{O}(N)$이다.

부분문제 6

이제 추가 제약이 없다.

등장하는 서로 다른 값들을 오름차순으로 $v_1 < v_2 < \cdots < v_M$이라고 하자.

각 값 v_i에 대해, L_i를 v_i가 처음 등장하는 위치, R_i를 v_i가 마지막으로 등장하는 위치라고 하자.

인접한 두 등장값 v_i, v_{i+1}를 생각하자.

만약 $R_i > L_{i+1}$이라면, 어떤 v_i가 어떤 v_{i+1}보다 뒤에 등장한다는 뜻이다. 즉, 원래 수열 안에 v_{i+1}, v_i 순서로 등장하는 두 원소가 존재한다.

이 둘이 같은 값 구간에 남아 있다면, 구간 내부의 상대적인 순서는 바뀌지 않으므로 비내림차순이 될 수 없다. 따라서 이 경우에는 반드시 v_i와 v_{i+1} 사이를 나누어야 한다.

이때 기준값 x는 $v_i \le x < v_{i+1}$가 되도록 고르면 된다. 예를 들어 $x = v_i$를 고를 수 있다.

반대로 $R_i < L_{i+1}$이라면, 모든 v_i가 모든 v_{i+1}보다 앞에 등장한다. 따라서 v_i와 v_{i+1}를 같은 구간에 두어도 이 둘 사이의 순서는 문제가 없다.

그러므로 반드시 나누어야 하는 경계는 정확히 $R_i > L_{i+1}$인 경계들이다.

이 경계들을 모두 나누면, 각 값 구간 안에서는 인접한 등장값들 v_i, v_{i+1}가 모두 $R_i < L_{i+1}$을 만족한다. 따라서 구간 안의 값들은 원래 수열에서도 작은 값부터 큰 값 순서로 등장하고, 그 구간의 부분수열은 비내림차순이다.

구간들 자체도 값의 범위가 증가하는 순서로 이어 붙여지므로, 전체 수열 역시 비내림차순이 된다.

따라서 정답은 $R_i > L_{i+1}$인 i의 개수이다.

구현은 다음과 같이 하면 된다.

1. 각 값 x에 대해 첫 등장 위치 $\text{first}[x]$, 마지막 등장 위치 $\text{last}[x]$를 계산한다.
2. 값 $1, 2, \ldots, N$을 순서대로 보면서 실제로 등장한 값만 확인한다.
3. 직전에 등장한 값이 p, 현재 등장한 값이 x일 때, $\text{last}[p] > \text{first}[x]$이면 답을 1 증가시킨다.

전체 시간 복잡도는 $\mathcal{O}(N)$이고, 메모리 복잡도는 $\mathcal{O}(N)$이다.