"평균 최대화 (average)" 문제 풀이

작성자: 조승현

부분문제 1

수열의 가능한 총 상태는 2^N개를 넘지 않는다. 각 상태를 정점으로, 들어내기 연산을 방향성 간선으로 생각하여 시작점으로부터 도달 가능한 정점을 모두 탐색하면 해결할 수 있다.

부분문제 2

$D[i][j][k]$: 초기 수열이 $A[i], \ldots, A[j]$일 때 총 k개의 원소가 제거된 경우 제거된 원소들의 합의 최솟값.

위와 같이 정의하면 (i, j)에서 들어내기 연산을 하는 경우는 따로 처리하고, 그 외의 경우에는 i와 j 사이에 남아있는 원소가 있을 것이므로 모든 $i < t < j$에 대해 $D[i][t]$와 $D[t][j]$로부터 업데이트해주면 $O(N^5)$ 시간에 문제를 해결할 수 있다.

부분문제 3

부분문제 2의 풀이에서 모든 $i < t < j$에 대해 갱신을 해줄 필요가 없고, $A[i+1], \ldots, A[j-1]$ 중 최솟값이 $A[t]$인 t에 대해서만 갱신해주어도 충분하다. 따라서, $O(N^4)$시간에 문제를 해결할 수 있다. 이 때 상수가 매우 작기 때문에 $N = 250$인 경우에도 빠른 시간에 동작한다.

부분문제 4

모든 $A[i]$가 4를 넘지 않을 때, 4만 들어내는 경우 평균이 증가할 수 없으므로 무조건 손해이고, 문제에서 주어지는 구간이 막힌 수열이기 때문에 2를 들어내려면 1만 남게 되어 평균이 증가하지 않는다. 따라서, 3과 4를 들어내는 방법들만 고려하면 충분하다. 이러한 들어내기 방법들은 서로 겹치지 않는 구간들이며, 이 유효한 구간들을 평균이 작은 순서대로 제거하는 것이 항상 최적이다.

부분문제 5

막힌 수열인 부분수열은 최대 $2N$개이고, 이 부분수열들은 서로 교차하지 않는다. 즉, 구간으로 표현했을 때 서로 포함관계이거나 겹치는 부분이 없다. $A[i], \ldots, A[j]$가 막힌 수열인 구간 $[i, j]$들을 포함 관계에 따라 트리 구조로 표현할 수 있다. 부분문제 2, 3의 풀이에서 나왔던 다이나믹 프로그래밍을 그대로 사용하여, $D[i][j]$를 노드 i에 해당하는 구간에서 j개 지웠을 때 그 합의 최솟값으로 놓으면 다이나믹 프로그래밍으로 $O(N^2)$ 시간에 문제를 해결할 수 있다.

부분문제 6

다이나믹 프로그래밍으로 $D[i][j]$를 계산하는 대신에, 노드 i에 해당하는 구간에서 x개를 제거했을 때 합 y로 가능한 경우를 모두 좌표평면 상의 점 (x, y)에 나타내고, 이의 convex hull을 생각하자. 부모의 convex hull은 자식의 convex hull의 민코프스키 합이 되고, 해당 노드의 구간 자체에 들어내기 연산을 했을 때에 해당하는 점 하나를 추가하는 식으로 업데이트가 된다. convex hull로 설명을 했지만, 실제로는 lower hull만 있으면 되기 때문에 기울기 기준의 priority queue로 저장하면서 small-to-large 트릭으로 관리해주면 된다. 이를 통해 $O(N \log^2 N)$ 시간에 문제를 해결할 수 있다.

부분문제 7

$A[i] \leq 20$이면 tree의 depth가 20보다 작아진다. 이를 이용하면 부분문제 5의 DP를 했을 때 총 계산량이 $O(N \times 20^2)$를 넘지 않는다.

부분문제 8

답을 구해야 하는 구간이 여러 가지이면 priority queue나 set 내에서 구간 합 연산을 해야 한다는 문제가 발생한다. 간단하게는 balanced binary search tree를 통해 구현하면 이를 해결할 수 있다. BBST를 쓰지 않고도 세그먼트 트리를 통해 문제를 해결할 수 있는데, 처음에 priority queue를 small to large trick을 통해 계산해준 다음, priority queue에 들어가는 모든 원소를 정렬 후 리넘버링하여 다시 동일한 과정을 해주면 이제 세그먼트 트리를 이용해 $O(N \log^2 N)$ 시간에 문제를 해결할 수 있다.