"중등부 4번 / 고등부 3번. 트리 뽑아내기" 문제 풀이

작성자: 이온조

부분문제 1

문제에 주어진 연산을 그대로 구현하면 $O(N^2)$의 시간복잡도로 문제를 해결할 수 있다.

부분문제 2

최솟값, 최댓값을 관리하는 자료구조인 힙(Heap)과 같은 원리로 뽑아내기 연산 이후에도 모든 $2 \le i \le N$에 대해 $A_{P_i} < A_i$가 성립한다는 것을 증명할 수 있다. 그러므로 루트 정점은 항상 모든 정점들 중 가장 작은 가중치를 갖게 된다. 그렇기 때문에 답이 항상 $1, 2, \cdots, N$이다.

부분문제 3

어떤 정점 u의 자식들 $v_1, \cdots, v_k$가 있을 때, 일반성을 잃지 않고 $A_{v_1} < \cdots < A_{v_k}$라고 하자. u가 특별한 경로로 선택된다면 가장 작은 가중치를 가진 v_1도 특별한 경로로 선택될 것이다. 부분문제 조건에 의해 뽑아내기 연산 이후에 v_1 정점의 가중치는 더욱 작아질 것이므로, 여전히 v_1이 특별한 경로로 선택됨을 알 수 있다. 즉, v_1을 루트로 하는 서브트리에 속한 정점들이 모두 제거될 때까지 v_1이 특별한 경로로 선택된다. 즉, $v_1, \cdots, v_k$ 순으로 그 정점의 서브트리에 있는 모든 정점들이 차례대로 선택된다. 그러므로 루트 정점에서부터 DFS를 수행하되 가중치가 작은 정점부터 방문하면 DFS에서의 정점 방문 순서가 답이라는 것을 알 수 있다.

부분문제 4

차수가 3 이상이거나 리프인 정점들만 고려하면 압축된 형태의 트리를 만들 수 있다. 이렇게 만든 압축된 트리의 간선에는 여러 개의 정점들이 줄지어 달려 있는 형태가 될 것이다. 압축된 트리의 각 정점마다 인접한 정점들의 가중치를 우선순위 큐로 관리하고, 압축된 트리의 간선에 줄지어 달려있는 정점들은 큐로 관리하자. 문제 조건에 의해 압축된 트리의 높이가 작으므로 특별한 경로를 $O(20 \log N)$의 시간복잡도로 찾을 수 있다. 특별한 경로를 따라 올라가면서 간선의 큐와 정점의 우선순위 큐를 갱신해주면 뽑아내기 연산 역시 $O(20 \log N)$ 시간복잡도로 구현할 수 있다. 총 시간복잡도는 $O(20N \log N)$이다.

부분문제 5

서브트리에서 부분문제를 해결하고, 이를 합쳐서 부모에 대한 문제를 해결하는 방식으로 해결하자. 어떤 정점 u의 자식들 $v_1, \cdots, v_k$가 있을 때, v_i를 루트로 하는 서브트리의 답이 수열 $B_{v_i}[\cdots]$라고 하자. 이는 v_i가 특별한 경로에 포함되어 뽑아내기 연산을 수행했을 때 v_i 정점의 가중치가 수열 $B_{v_i}[\cdots]$ 순서대로 변화한다는 의미이다. 우리는 이 정보를 이용해서 u를 루트로 하는 서브트리의 답 수열 $B_u[\cdots]$를 구해야 한다. 자명히 $B_u[\cdots]$의 첫 번째 수는 A_u이다. 이제 나머지 수들은 병합 정렬을 수행하듯이 $B_{v_1}[\cdots], \cdots, B_{v_k}[\cdots]$ 중 가장 작은 수를 뽑아 $B_u[\cdots]$의 끝에 추가해주는 것을 반복하면 모두 구할 수 있다.

이를 단순히 구현하면 시간 초과를 받기 때문에 최적화가 필요하다. 수열 $B_i[\cdots]$을 앞에서부터 봤을 때 최댓값이 갱신되는 순간 새로운 그룹으로 묶는 방식으로 그룹을 나누어 관리하고, 그룹의 대푯값을 그룹의 맨 왼쪽 수로 정의하자. 예를 들어 $2, 1, 3, 5, 4, 8, 6, 7$은 $[\mathbf{2}, 1]$, $[\mathbf{3}]$, $[\mathbf{5}, 4]$, $[\mathbf{8}, 6, 7]$로 관리하는 것이다. 이렇게 그룹을 나누면 여러 수열을 하나로 합칠 때 그룹으로 묶인 수들은 항상 그룹으로 묶여 있고, 그룹의 대푯값

순서대로 합쳐진다는 것을 알 수 있다. 또한 첫 번째 수보다 대푯값이 작은 그룹들은 하나의 새로운 그룹으로 묶이게 된다. 그러므로 그룹들을 우선순위 큐로 관리하고, 작은 집합에서 큰 집합으로 합치는 테크닉을 활용하면 $O(N \log^2 N)$의 시간복잡도로 문제를 해결할 수 있다.

조금 더 생각해보면, 결국 이 풀이에서 구하는 것이 부모-자식 순서 관계가 유지되는 순열 중 사전순으로 가장 앞서는 순열이라는 것을 귀납적으로 증명할 수 있다. 그러므로 우선순위 큐를 사용해서 루트 정점에서부터 가장 가중치가 작은 정점을 선택해나가는 방식으로 $O(N \log N)$에 문제를 해결할 수 있다.