정렬하기 – 풀이

작성자: 박영우

물건을 정점으로, 실을 간선으로 생각하면 한 질문(쿼리)의 반환값은 트리에서의 독립 집합(independent set) 중 가중치가 가장 큰 것이다.

다음과 같은 방법으로 어떤 두 정점 a, b에 대해 $A[a] \le A[b]$임을 확인할 수 있다.

세 점 u, v, x를 고르고 나머지 정점들의 집합을 S라 하자. u, v, x를 순서대로 직선으로 연결하고 S에 속한 정점을 전부 x에 연결한 후 쿼리를 호출한다.

쿼리의 반환값에서 x가 선택된 경우와 그렇지 않은 경우로 나눌 수 있다.

- x가 선택되지 않은 경우: $A[u], A[v]$ 중 하나가 0이 아니라면 $A[u], A[v]$ 중 작지 않은 값이 선택되었을 것이므로 $A[u] \le A[v]$ 혹은 $A[v] \le A[u]$임을 확인할 수 있다. u, v중 아무 정점도 선택되지 않은 경우 $A[u] = A[v] = 0$임을 알 수 있다.
- x가 선택된 경우: S에 있는 정점들의 A값의 합보다 $A[x]$가 더 작다면 x를 선택하지 않고 S에 있는 정점을 선택하는 것이 이득이므로 S의 임의의 정점 w에 대해 $A[x] \ge A[w]$이다.

비슷하게, $A[u] \le A[v]$임을 아는 쌍이 있다면 u, v를 포함하지 않고 서로 겹치지 않는 순서쌍 $(x_1, y_1), \cdots, (x_k, y_k)$가 있을 때 각각의 i에 대해 $A[x_i] \le A[y_i]$이거나 $A[x_i] \ge A[y_i]$ 중 하나를 알 수 있다. x_i, y_i, u를 순서대로 연결하고, u와 v를 연결하고, 위 과정에서 어느 정점과도 연결되지 않은 모든 정점을 u와 연결한 후 쿼리를 호출하면 된다.

u가 선택된 경우와 그렇지 않은 경우로 나눌 수 있다.

- u가 선택되지 않은 경우, 앞선 논리와 비슷하게 x_i와 y_i 중 하나가 선택되었다면 $A[x_i] \le A[y_i]$, $A[x_i] \ge A[y_i]$ 중 하나를 알 수 있고, 둘 다 선택되지 않았다면 $A[x_i] = A[y_i] = 0$이다.
- u가 선택된 경우 $A[u] \le A[v]$이므로 반환된 집합에서 u를 제외시키고 v를 추가한 집합도 최대 독립 집합이다. 따라서 이 집합에서 이전 논리를 적용하면 $A[x_i] \le A[y_i]$, $A[x_i] \ge A[y_i]$ 중 하나를 알 수 있다.

위의 방법을 이용하여 처음 $A[u] \le A[v]$임을 알아낸 u, v를 제외한 $N-2$개의 원소를 분할 정복을 통해 수열을 정렬할 수 있다. 수열을 두 부분으로 분할하면 각각을 정렬하는 것은 병렬적으로 진행할 수 있다. 길이가 같은 정렬된 두 수열을 하나의 정렬된 수열로 병합하는 것은 다음과 같이 할 수 있다.

- 길이 $2N$이고 앞쪽 절반과 뒤쪽 절반이 각각 정렬되어있는 배열 A를 다음과 같이 정렬할 수 있다.
  N은 짝수이며 A의 모든 원소가 서로 다르다고 가정할 수 있다.
  우선 A의 홀수번째 인덱스, 짝수번째 인덱스를 따로 모아 이 인덱스들끼리 정렬한다.
  이렇게 하여 만들어진 배열을 A', A를 정렬하면 얻을 수 있는 배열을 A''이라 하자.
  A'의 홀수 번째 인덱스 $A'[2k+1]$에 대해 생각하자. 일반성을 잃지 않고 이 인덱스가 A의 앞쪽 절반에 위치했다 하자.

A의 홀수번째 원소들 중 $A'[2k+1]$보다 작은 것은 총 k개 존재하는데, 배열 앞쪽 절반에서 a개, 뒤쪽 절반에서 b개 존재한다 하자.
그러면 배열 앞쪽 절반에서는 $A'[2k+1]$보다 작은 원소가 $2a+1$개($A'[2k+1]=A[2a+1]$이므로), 뒤쪽 절반에서는 $2b+1, 2b+2$개 중 하나이다. ($A[2b+1] < A'[2k+1] < A[2b+3]$)
따라서 $A'[2k+1]$은 $A''[2k+1], A''[2k+2]$중 하나이다.
마찬가지로 짝수 번째 인덱스 $A'[2k]$는 $A''[2k]$나 $A''[2k-1]$중 하나이다.
따라서 A'에서 모든 $2k+1, 2k+2$ 쌍에 대해 각각 정렬해주면 A''을 얻을 수 있다.

크기 2^k의 두 수열을 합치는 것을 $k+1$회의 연산으로 할 수 있고, 따라서 2^k 크기의 수열을 정렬하는 것은 $k(k+1)/2$회에 할 수 있다. 이렇게 하면 $N-2$개의 값을 정렬하는 것을 55회에 할 수 있다.

또한, 길이가 $N-2$인 정렬된 배열에 처음에 구한 대소 관계를 아는 2개의 수 u, v를 끼워넣는 과정은 다음과 같이 할 수 있다.

N이 10 이하인 경우 u, v를 제외하고 대소관계를 아는 수 세 개가 있어 ($5 \le N$), u와 다른 수를 비교하는 것이 가능하므로 u, v와 나머지 $N-2$개의 쌍의 대소관계를 전부 비교해 보는 방법으로 풀 수 있다.

N이 큰 경우에는 분할 정복을 사용한다. 배열의 가운데에 있는 수를 x, y라 할 때, u, x와 y, v를 동시에 비교한다. $A[u] \le A[v], A[x] \le A[y]$이므로 결과는 다음 중 하나이다.

- $A[u] \le A[x], A[y] \le A[v]$: 이 경우 u가 앞쪽 절반에 들어가고, v가 뒤쪽 절반에 들어가므로 이분 탐색을 통해 찾을 수 있다. 이때 두 이분 탐색을 병렬적으로 시행할 수 있다.
- $A[x] \le A[u], A[v] \le A[y]$: $A[x] \le A[u] \le A[v] \le A[y]$임을 알 수 있으므로 x, y 사이에 u, v가 들어가게 된다.
- $A[u] \le A[x], A[v] \le A[y]$: 두 수 모두 앞쪽 절반에 위치한다.
- $A[x] \le A[u], A[y] \le A[v]$: 두 수 모두 뒤쪽 절반에 위치한다.

따라서 전체 $O(\lg N)$ 쿼리에 해결할 수 있고, 최악의 상황에서도 12회 이하의 쿼리를 사용함을 보일 수 있다.

따라서 사용하는 총 쿼리의 횟수는 최대 $1+55+12=68$회로, 주어진 제한 조건을 충족한다.