

점프 - 풀이

작성자: 박영우

부분문제 1

주어진 N 개의 발판을 정점으로 생각하고 i 번 발판에서 j 번 발판으로 이동할 수 있는 모든 (i, j) 쌍에 대해 간선을 추가하여 방향그래프를 구성할 수 있다. 문제에서 구해야 하는 값은 각 정점에서 도달할 수 있는 정점의 개수이다.

각 정점의 쌍 (i, j) 에 대해 i 에서 j 로 향하는 간선이 있는지 $\mathcal{O}(1)$ 에 판별할 수 있다. 따라서 그래프를 $\mathcal{O}(N^2)$ 에 구성할 수 있고, 각 정점에 대해 해당 정점에서 도달할 수 있는 정점의 집합을 DFS 등의 그래프 탐색 알고리즘을 사용해 $\mathcal{O}(N^2)$ 에 구할 수 있다. 따라서 전체 시간복잡도는 $\mathcal{O}(N^3)$ 이다.

부분문제 2

i 번 발판에서 도달 가능한 발판들의 집합을 다음과 같이 분류한다.

- L_i : $X_j < X_i$ 이며 i 번 발판에서 도달 가능한 j 번 발판들의 집합
- M_i : $X_i = X_j$ 이며 i 번 발판에서 도달 가능한 j 번 발판들의 집합
- R_i : $X_i < X_j$ 이며 i 번 발판에서 도달 가능한 j 번 발판들의 집합

어떤 i 번 발판에 대해 $R_i \neq \emptyset$ 이라 하자. 이때 k 를 R_i 의 최소 원소라 하자.

임의의 원소 $j \in R_i$ 에 대하여 다음 두 가지 경우 중 정확히 하나만 참이다.

- $X_j > X_k$: i 에서 j 로 가는 경로 중 처음으로 x 좌표가 X_k 를 초과하는 발판을 v , 직전 발판을 u 라 하자. 즉, $X_u \leq X_k < X_v$ 이다.
이때 $X_v - X_u \leq D$ 이므로 $X_v - X_k \leq D$ 이다. 또한 $v \in R_i$ 이므로 k 의 최소성에 의해 $k < v$ 이다. 따라서 k 에서 v 로 직접 이동이 가능하므로 k 에서 v 를 거쳐 j 로 가는 경로가 존재하고, $j \in R_k$ 이다.
반대로 $j \in R_k$ 를 가정하면, R_k 의 모든 원소는 k 를 거쳐 i 에서 도달 가능하므로 $j \in R_i$ 이며, 정의에 의해 $X_k < X_j$ 이다.
따라서 이 경우에서 고려되는 j 들의 집합은 정확히 R_k 이다.
- $X_j \leq X_k$: $X_k \leq X_i + D$ 이므로, $X_i < X_j \leq X_k$ 이면 $X_j - X_i \leq D$ 를 만족한다. 조건에 의해 $i < j$ 이므로, 이를 만족하는 j 는 모두 i 에서 한 번에 이동 가능하다.

따라서 다음 관찰이 성립한다.

R_i 의 각 원소는 다음 중 정확히 하나에 속한다.

- R_k
- $i < j, X_i < X_j \leq X_k$ 인 j 들의 집합

마찬가지로 L_i 에 대해서도 다음과 같은 관찰을 할 수 있다.

$L_i \neq \emptyset$ 일 때, k 를 L_i 의 원소 중 최소 원소라 하자. 그러면 L_i 의 각 원소는 다음 중 정확히 하나에 속한다.

- L_k
- $i < j, X_k \leq X_j < X_i$ 인 j 들의 집합

위 관찰을 통해 각 $i = N, N - 1, \dots, 1$ 에 대해 순서대로 R_i, L_i, M_i 를 $\mathcal{O}(N)$ 시간에 구할 수 있다. 따라서 전체 $\mathcal{O}(N^2)$ 시간복잡도에 문제를 해결할 수 있다.

부분문제 3

i 번 발판에서 j 번 발판에 도달할 수 있다면 $i, i + 1, \dots, j$ 번 발판 순서대로 이동하는 경로가 존재한다. 따라서 다음과 같이 두 가지의 경우로 나눌 수 있다.

- i 번 발판에서 $i + 1$ 번 발판으로 이동할 수 있는 경우: 이 경우 i 번 발판에서 도달할 수 있는 발판은 $i + 1$ 번 발판에서 도달할 수 있는 발판들과 i 번 발판 자신이다.
- i 번 발판에서 $i + 1$ 번 발판으로 이동할 수 없는 경우: 이 경우 i 번 발판에서는 다른 어떤 발판으로도 이동할 수 없다.

따라서 $N, N - 1, \dots, 1$ 번 발판에 대해 순서대로 답을 구해줄 수 있다. 전체 시간복잡도는 $\mathcal{O}(N)$ 이다.

부분문제 4

$X = \max\{X_1, X_2, \dots, X_N\}$ 이라 하자.

S_x 를 $X_i = x$ 인 i 번 발판들의 집합이라 하자.

모든 S_x 를 $\mathcal{O}(N + X)$ 에 구할 수 있다. 각 i 에 대해 R_i 의 최소 원소 k 는 $S_{X_{i+1}}, \dots, S_{\min\{X, X_i + D\}}$ 의 원소 중 i 초과인 최소 원소이다. 각 집합 S_x 에 대해 i 를 초과하는 가장 작은 원소는 이분 탐색을 통해 $\mathcal{O}(\log N)$ 에 찾을 수 있고, 따라서 k 를 $\mathcal{O}(X \log N)$ 에 찾을 수 있다.

또한 $X_i < X_j \leq X_k$ 이고 $i < j$ 인 j 들의 개수도 같은 방법으로 $\mathcal{O}(X \log N)$ 에 찾을 수 있다. 따라서 각 $i = N, N - 1, \dots, 1$ 에 대해 R_i 의 크기를 순서대로 $\mathcal{O}(X \log N)$ 에 구할 수 있다.

같은 방법으로 L_i 의 크기도 같은 시간복잡도에 구할 수 있다.

M_i 의 경우 $i \leq j, X_i = X_j$ 인 발판의 수를 세면 된다. 즉, M_i 의 크기는 S_{X_i} 에서 i 이상인 원소의 개수와 같다. 이는 이분 탐색을 통해 $\mathcal{O}(\log N)$ 에 찾을 수 있다.

따라서 전체 시간복잡도는 $\mathcal{O}(NX \log N)$ 이다.

부분문제 5

S_x 를 부분문제 4와 같이 정의한다. 그렇다면 각 i 에 대해 R_i 의 최소 원소 k 는 $S_{X_{i+1}}$ 의 원소 중 i 초과인 최소 원소이다. 주어진 인덱스 범위가 크므로 모든 S_x 를 `std::map` 등의 자료 구조로 관리하면 k 를 $\mathcal{O}(\log N)$ 에 구할 수 있다.

또한 어떤 j 번 발판이 $X_i < X_j \leq X_k, i < j$ 를 만족한다면 $X_j = X_k$ 이다. 이러한 j 의 개수도 $\mathcal{O}(\log N)$ 에 셀 수 있다.

따라서 각 $i = N, N - 1, \dots, 1$ 에 대해 R_i 의 크기를 순서대로 $\mathcal{O}(\log N)$ 에 구할 수 있다.

비슷한 방법으로 L_i 와 M_i 의 크기도 같은 시간복잡도에 구할 수 있고, 전체 시간복잡도는 $\mathcal{O}(N \log N)$ 이다.

부분문제 6

$i = N, N - 1, \dots, 1$ 순서대로 R_i 의 크기를 구할 수 있다.

- i 번 발판에 대해 R_i 의 최소 원소 k 는 현재까지 고려된 발판 ($i + 1, \dots, N$ 번 발판)들 중 x 좌표가 X_i 초과, $X_i + D$ 이하인 발판들 중 인덱스가 가장 작은 것이다. 따라서 이를 세그먼트 트리를 통해 관리하면 $\mathcal{O}(\log N)$ 시간에 구할 수 있다. 발판들의 x 좌표 범위가 크므로, 좌표 압축 등의 방법이 필요하다.
- R_i 는 R_k 와 $X_i < X_j \leq X_k, i < j$ 를 만족하는 발판들로 구성되어 있다. 두 번째 종류에 해당하는 발판의 개수는 현재까지 고려된 발판 중 x 좌표가 X_i 초과, X_k 이하인 발판들의 개수와 같다. 이 값 역시 세그먼트 트리를 통해 현재까지 고려된 발판들의 집합을 관리하면 $\mathcal{O}(\log N)$ 시간에 구할 수 있다.

L_i 의 크기는 같은 방법으로 구할 수 있고, M_i 의 크기는 부분문제 5의 방법을 사용하면 같은 시간복잡도에 구할 수 있다.

따라서 전체 시간복잡도는 $\mathcal{O}(N \log N)$ 이다.