

게임 - 풀이

작성자: 박영우

각 j ($1 \leq j \leq Q$)에 대하여, j 번째 게임에서 s_j 번 방에 출구가 있는 경우 앨리스가 승리한다. 이러한 경우를 먼저 처리한다. 앞으로의 서술에서는 s_j 번 방에 출구가 없다는 것을 가정한다.

부분문제 1

각 j ($1 \leq j \leq Q$)에 대하여, j 번째 게임에서 $s_j = 1$ 일 경우 앨리스가 승리한다. 그렇지 않을 경우 앨리스가 승리할 조건은 다음과 같다.

- $c_1, c_2, \dots, c_{s_j-1}$ 의 최솟값이 k_j 보다 크거나 같다.

위 조건이 성립하면, 앨리스는 1번 방에 도착하기 전까지 현재 방에서 번호가 정확히 1 줄어드는 방으로 향하는 통로를 k_j 개 선택할 수 있다. 따라서 앨리스가 승리한다.

그렇지 않을 경우, $c_t < k_j$ 이며 $1 \leq t < s_j$ 인 정수 t 가 존재한다. 이 경우 밥은 앨리스가 $t+1$ 번 방에 도달하기 전까지는 아무 통로를 선택하다가, $t+1$ 번 방에 도달했을 때 $t+2$ 번 방으로 향하는 통로를 선택하면 된다. (만약 $t+1 = N$ 이라면, 앨리스는 다른 방으로 이동할 수 없으므로 승리할 수 없다) 그러면 앨리스는 t 번 방으로 이동할 수 없고, s_j 번 방에서 시작하여 1번 방에 도달하기 위해서는 반드시 t 번 방을 거쳐야 하므로 앨리스는 승리할 수 없다.

$i = 1, 2, \dots, N-1$ 에 대해 $\min(c_1, c_2, \dots, c_i)$ 를 $\mathcal{O}(N)$ 에 전처리해 두면 각각의 게임에 대해 앨리스가 승리하는지 $\mathcal{O}(1)$ 에 판정할 수 있다.

부분문제 2

이 부분문제에서 1번 방은 다른 모든 방과 통로로 직접 연결되어 있고, 1번 방이 아닌 다른 두 방을 연결하는 통로는 없다.

1번 방에 출구가 있는 경우와 없는 경우로 나눠서 해결한다.

1번 방에 출구가 있는 경우, 각 게임에서 앨리스가 승리할 수 있다면 1번 방에 도달할 수 있다. 따라서 각 게임에 대해 1번 방에 도달할 수 있는지만 검사해도 충분하다.

1번 방에 출구가 없었다 하자. j 번째 게임에 대해 앨리스가 출구가 있는 방에 도달할 수 있다면, 우선 1번 방에 도달할 수 있어야 한다. 우선 이 조건을 검사하고, 성립하지 않는다면 앨리스가 패배한다고 판정한다.

1번 정점에 앨리스가 있고, 이 시점에서 앨리스가 승리할 수 있다면 정확히 한 번의 이동을 통해 출구가 있는 방에 도달할 수 있어야 한다. 그렇지 않다면, 앨리스는 1번 방이 아닌 다른 방으로 오게 되고, 앨리스는 다시 1번 방으로 돌아와야 한다. (혹은 해당 방에 연결된 통로의 수가 적어 더 이상 이동할 수 없다) 밥은 통로를 잘 선택하여 앨리스가 1번 방과 출구가 없는 다른 방을 무한히 왕복하도록 하게 할 수 있고, 따라서 앨리스는 승리할 수 없다.

따라서 1번 방에서 정확히 한 번의 이동을 통해 출구가 있는 방으로 이동할 수 있는지 판정하면 된다. 앨리스가 출구가 없는 방으로 향하는 통로를 고른다면, 밥은 그 통로를 선택하여 앨리스가 출구가 없는 방으로 이동하게 할 수 있다. 따라서 앨리스는 1번 방에 있을 때 출구가 있는 방으로 향하는 통로만을 골라야 한다. j 번째 게임에서 앨리스가 1번 방에 왔을 때, k_j 가 그러한 통로의 수보다 많다면 승리할 수 없고, 그렇지 않다면 승리할 수 있다.

부분문제 3

각 게임에 대해 $k_j = 1$ 이므로, 앨리스는 각 게임에서 정확히 하나의 통로를 고른다. 따라서 밥의 선택과 무관하게 앨리스는 매 이동에서 원하는 인접한 방으로 이동할 수 있다.

따라서, 앨리스가 j 번 게임에서 이길 필요충분조건은 s_j 번 방과 출구가 있는 방이 통로를 통해 연결되어 있는 것이다.

방과 통로들을 그래프로 해석하면, 출구가 있는 방과 s_j 번 방이 같은 연결 요소(connected component)에 있는 것이다.

처음에 주어진 방들을 dfs나 Union-Find 등의 알고리즘을 사용해 같은 연결 요소에 속한 방끼리 분류하고, 각 연결 요소에 출구가 있는지 구하면 쿼리를 $\mathcal{O}(1)$ 의 시간복잡도에 처리할 수 있다.

전체 시간복잡도는 $\mathcal{O}(N + M + Q)$, Union-Find를 사용할 경우 $\mathcal{O}((N + M + Q)\alpha(N))$ 이다.

부분문제 4

양의 정수 k 에 대해 V_k 를 다음과 같이 정의한다.

- 앨리스가 게임에서 k 개의 통로를 선택할 수 있다고 할 때, 앨리스가 게임에서 승리하게 되는 시작 방 번호의 집합은 V_k 이다.

즉, 각 j 에 대해 $s_j \in V_k$ 라면 앨리스가 승리하고, 그렇지 않으면 패배한다.

양의 정수 k 에 대해, 다음과 같은 관찰을 할 수 있다.

- v 번 방에 출구가 있다면 $v \in V_k$ 이다.
- v 와 연결된 통로 중, V_k 에 있는 방으로 향하는 통로가 k 개 이상이라 하자. 그렇다면 $v \in V_k$ 이다.

따라서, V_k 를 구하기 위해 다음과 같은 알고리즘을 생각할 수 있다.

1. 집합 S 를 공집합으로 초기화한다. 각 방 v 에 대해, d_v 를 v 와 인접한 통로 중 S 에 있는 방으로 향하는 통로의 개수로 정의한다. 처음에 d_v 는 모두 0이다.
2. 출구가 있는 방 v 에 대하여 v 를 S 에 추가한다. 이때 v 와 c 개의 통로로 인접한 방 u 에 대해 d_u 에 c 가 더해진다. v 와 인접한 모든 방 u 에 대해 d_u 를 갱신한다.
3. 만약 어떤 방 v 에 대해 v 가 S 에 없고, $d_v \geq k$ 라면 v 를 S 에 추가한다. 전과 같은 규칙으로 v 와 인접한 방 u 들에 대해 d_u 를 갱신한다.
4. 이 과정을 S 에 없는 모든 방 v 에 대해 $d_v < k$ 가 될 때까지 반복한다.

두 번째 과정이 끝난 이후, S 는 출구가 있는 방의 집합이다. 따라서 이 시점에서 $S \subseteq V_k$ 이다.

세 번째 과정에서 S 에 추가된 모든 v 에 대해 $d_v \geq k$ 이다. 즉, v 와 연결된 통로 중 v 가 추가되기 직전 시점의 S 에 속하는 방과 연결된 것이 k 개 이상이다. 따라서 v 를 추가하기 직전 $S \subseteq V_k$ 였다면, v 를 추가한 이후에도 $S \subseteq V_k$ 이다.

따라서 위 알고리즘이 종료되었을 때, $S \subseteq V_k$ 이다.

이제 $V_k \subseteq S$ 를 증명한다. S 에 없는 방 v 가 V_k 에도 없음을 보이면 충분한데, 이는 S 에 없는 방 v 에 대하여 앨리스가 v 에서 시작했을 때 승리할 수 없음을 의미한다. 이는 다음과 같이 증명할 수 있다.

- v 번 방과 연결된 통로가 k 개 미만일 경우, 앨리스는 다른 방으로 이동할 수 없는데 v 번 방에는 출구가 없으므로($v \notin S$) 앨리스가 승리할 수 없다.
- 그렇지 않을 경우, v 번 방과 연결된 통로 중 S 에 속하는 방으로 향하는 것은 k 개 미만이다. 따라서, 앨리스가 어떻게 통로를 선택하더라도, 그 통로 중 적어도 하나는 S 에 없는 방으로 향한다.
- 밥이 그러한 통로를 반복해서 선택하는 전략을 사용할 경우, 앨리스가 어떻게 이동하더라도 S 에 속한 방으로 이동할 수 없다. 출구가 있는 모든 방은 S 에 속하므로, 앨리스는 출구가 있는 방에 도달할 수 없어 승리할 수 없다.

따라서, $V_k = S$ 이다.

이 부분 문제에서는 모든 게임에 대해 k_j 가 동일하므로, 한 V_k 만 구하면 충분하다. 앞서 제시한 알고리즘을 그대로 사용할 수 있는데, d_u 가 k 이상이며 S 에 속하지 않는 방 u 들을 큐(queue) 등의 자료구조를 사용하여 관리하면 충분하다.

위 알고리즘에서 방 u 가 S 에 추가되었을 때, u 와 연결된 통로들이 한 번 순회된다. 따라서 각 $i(1 \leq i \leq M)$ 에 대해, 방 a_i 와 b_i 를 연결하는 통로들은 최대 두 번 확인된다.

전체 시간복잡도는 $\mathcal{O}(N + M + Q)$ 이다.

부분문제 5

이 부분문제에서는 모든 게임에 대해 앨리스가 시작하는 방 s_j 가 같다. $s_1 = s_2 = \dots = s_Q = s$ 라 하자.

다음과 같은 관찰을 할 수 있다.

- 모든 양의 정수 i 에 대하여, $V_{i+1} \subseteq V_i$ 이다.

부분문제 4의 알고리즘에서 S 에 정점이 추가되는 조건은 k 가 증가함에 따라 더 강해지는데, 따라서 $k = i + 1$ 일 때 알고리즘에서 v 가 S 에 추가되었다면 $k = i$ 일 때에도 추가된다.

따라서, $s \in V_k$ 인 최대 k 가 존재한다면, $1 \leq k' \leq k$ 인 모든 k' 에 대해 $s \in V_{k'}$ 이다.

부분문제 4의 알고리즘을 사용하여 한 k 에 대해 V_k 를 구할 수 있으므로, 이분 탐색으로 $s \in V_k$ 인 k 를 찾거나, 존재하지 않는지 판정할 수 있다.

이후 각 게임 j 에 대해서 k_j 와 k 의 대소를 비교하여 앨리스가 승리하는지 판정할 수 있다.

모든 게임에서 주어지는 k_j 중 최댓값을 X 라 하자. $k = 1, 2, \dots, X$ 범위에 대해서만 이분 탐색을 적용해보면 충분하므로, 전체 시간복잡도는 $\mathcal{O}((N + M) \log X + Q)$ 이다.

부분문제 6

이 부분문제에서 $N, M, k_j \leq 3000$ 이다. 따라서 $X \leq 3000$ 이다.

양의 정수 $k = 1, 2, \dots, X$ 에 대해 V_k 를 각각 구하면, 모든 게임에 대하여 앨리스가 승리하는지 $\mathcal{O}(1)$ 에 판정할 수 있다.

전체 시간복잡도는 $\mathcal{O}((N + M)X + Q)$ 이다.

부분문제 7

출구가 없는 각 방 v 에 대해, $v \in V_k$ 를 만족하는 최대 k 를 K_v 라 하자. 그러한 k 가 없다면 $K_v = 0$ 으로 둔다. 또한, v 에 출구가 있다면 $K_v = 10^{18}$ 로 둔다. 그러면 각 게임 j 에서 엘리스가 승리할 필요충분조건은 $k_j \leq K_{s_j}$ 이다.

모든 K_v 를 한 번에 구하기 위해 부분문제 4의 알고리즘을 k 가 큰 값에서 작은 값으로 감소시키며 실행한다.

현재까지 V_k 에 포함되었다고 판정한 방들의 집합을 S 라 하자. S 에 포함되지 않은 각 방 v 에 대해, d_v 를 v 와 S 에 속한 방들을 연결하는 통로의 수로 정의한다.

처음에는 출구가 있는 방들을 모두 S 에 넣고, 현재 k 를 10^{18} 로 둔다.

현재 k 에 대해 $d_v \geq k$ 인 방 v 가 있다면, 부분문제 4의 관찰에 따라 v 도 V_k 에 포함된다. 따라서 이러한 방을 S 에 추가하고, 인접한 방들의 d 값을 갱신한다. 새롭게 조건을 만족하는 방이 생길 수 있으므로, 그러한 방들을 큐를 사용하여 관리하고, 더 이상 생기지 않을 때까지 반복한다.

더 이상 S 에 추가할 방이 없다고 하자. 이때 S 에 속하지 않은 방들의 d_v 중 최댓값을 x 라 하면, $x < k' \leq k$ 인 모든 k' 에 대해 $V_{k'} = S$ 이다. 또한 $d_v = x$ 인 방들은 $k = x$ 일 때 처음으로 V_k 에 포함될 수 있다.

따라서 현재 S 에 속하지 않은 방 중 d_v 가 가장 큰 방을 찾아, 현재 k 를 그 값으로 낮추고 다시 같은 과정을 반복하면 된다. 이때 새롭게 S 에 추가되는 방 v 에 대해 $K_v = k$ 로 둔다.

S 에 속하지 않은 방 중 d_v 가 가장 큰 방을 빠르게 찾는 것은 우선순위 큐 등을 사용하여 처리할 수 있다. d_v 가 갱신될 때마다 (d_v, v) 를 우선순위 큐에 삽입하고, d_v 가 가장 큰 방을 찾을 때에는 우선순위 큐에서 가장 큰 값을 꺼내면 된다. 이때 이미 S 에 포함된 방에 대한 원소는 무시하고, v 가 S 에 포함되지 않는 (d_v, v) 가 나올 때까지 반복하면 된다.

$K_v = 0$ 인 방의 경우, d_v 가 한 번도 갱신되지 않으므로 위 과정에서 처리되지 않는다. 이러한 방들에 대해서는 따로 처리한다.

각 방은 S 에 한 번만 추가되고, 각 통로 정보는 양 끝 방이 추가될 때에만 필요하므로 최대 두 번 확인된다. 이로 인해 우선순위 큐에는 $\mathcal{O}(M)$ 개의 원소가 삽입된다.

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