"고등부 1번. 점프" 문제 풀이

작성자: 이종서

부분문제 1

처음 방문하는 정점이 1이고 마지막으로 방문하는 정점이 N일 때 가능한 정점의 방문 순서는 총 $(N-2)!$ 가지 있다. 이 방문 순서를 전부 순회하며 간선을 지난 횟수가 입력과 같아지는 것을 찾아 문제를 해결할 수 있다. 시간 복잡도는 $O(N^2 \times (N-2)!) = O(N!)$이다.

부분문제 2

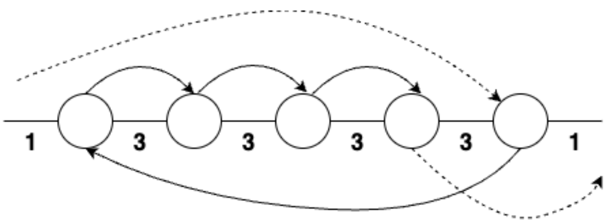
간선을 지난 횟수를 나타내는 수열 c에서는 다음과 같이 연속된 1과 연속된 3이 반복되어 나타나게 된다.

$$c = (\underbrace{1, 1, \ldots, 1}_{\text{연속된 1}}, \underbrace{3, 3, \ldots, 3}_{\text{연속된 3}}, \cdots, \underbrace{1, 1, \ldots, 1}_{\text{연속된 1}})$$

$c_i = 1$인 간선들은 정방향으로 한 번 지나야 하고, $c_i = 3$인 간선들은 정방향으로 두 번, 역방향으로 한 번 지나야 함을 관찰할 수 있다.

이를 기반으로 그래프의 간선들을 $c_i = 1$인 연속된 간선의 구간과 $c_i = 3$인 연속된 간선의 구간으로 나누어 생각할 수 있다. 구간을 앞에서부터 순회하며,

- $c_i = 1$ 구간에서는 간선들을 한 번만 지나면 되므로, 대응되는 정점들을 순방향(정점 번호가 증가하는 순서)으로 하나씩 방문한다.
- $c_i = 3$ 구간에서는 간선들을 역방향으로도 한 번 지나야 하므로, 구간의 마지막 정점을 먼저 방문한 후, 시작 정점으로 돌아와 나머지 정점을 차례로 하나씩 방문한다.



이를 구현하면 $O(N)$ 시간 복잡도에 해결할 수 있다. 구현시 두 구간의 경계에 걸쳐있는 정점에 대한 처리에 유의해야 한다.

부분문제 3

연속한 동일한 c 값을 갖는 간선들은 한 번에 연달아 통과(방문)해도 무방하다. 따라서

$$c_1 < c_2 < \cdots < c_M > c_{M+1} > \cdots > c_{N-1}$$

와 같이 입력이 강증가 후 강감소 형태라고 가정할 수 있다. 이때는 항상

$$|c_i - c_{i+1}| = 2$$

가 성립한다. 즉, c 수열이

$$(1, 3, \ldots, 2k-3, 2k-1, 2k-3, \ldots, 3, 1)$$

꼴이면 충분하다.

이 경우의 방문 순서는

$$p = (1, 2k-1, 2, 2k-2, 3, 2k-3, \ldots, k, 2k)$$

로 잡으면 모든 간선을 정확히 요구 횟수만큼 통과하면서 정점 1에서 $N = 2k$까지 도달할 수 있다.

이를 구현하면 $O(N)$ 시간에 해결할 수 있다.

부분문제 4

어떤 점프가 더 작은 번호의 정점에서 더 큰 번호의 정점으로 이동한다면 '순방향', 그렇지 않다면 '역방향'이라고 부르자. 2 이상 $N-1$ 이하의 정수 i에 대해 정점 i의 앞/뒤 간선을 지난 횟수의 차이를 $d_i := c_i - c_{i-1}$로 나타내자. 만약, $c_1 = c_{N-1} = 1$이고, d_i 조건을 만족하는 점프 방법을 찾는다면, 문제를 해결할 수 있다.

먼저, 다음 사실을 관찰할 수 있다:

- $d_i \in \{+2, 0, -2\}$
- $d_i = +2$인 정점 i는 직전/직후 방문 정점의 번호가 모두 i보다 크다. 다시 말해, $p_j = i$라면,

$$p_{j-1}, p_{j+1} > i$$

- $d_i = -2$인 정점 i는 직전/직후 방문 정점의 번호가 모두 i보다 작다. 다시 말해, $p_j = i$라면,

$$p_{j-1}, p_{j+1} < i$$

- $d_i = 0$인 정점 i는 직전/직후 방문 정점들의 i와의 대소 관계가 서로 다르다. 다시 말해, $p_j = i$라면,

$$\min(p_{j-1}, p_{j+1}) < i < \max(p_{j-1}, p_{j+1})$$

두 집합 $S := \{i \mid d_i = -2\}$와 $E := \{i \mid d_i = +2\}$를 정의하자. $S := \{s_1, s_2, \cdots\}$, $E := \{e_1, e_2, \cdots\}$ 와 같이 $s_1 \le s_2 \le \cdots$ 이고 $e_1 \le e_2 \le \cdots$ 이게 각 원소에 크기 순으로 번호를 붙이자. 이 때, $c_i - c_1 = \sum_{t=1}^{i-1} d_t$로 부터 다음 두 사실을 알 수 있다:

- $c_{n-1} = 1$이므로, $\sum d_t = 0$이 되어 $|S| = |E|$.
- 모든 i에 대해 $c_i \ge 1$이므로, 항상 $\sum_{t=1}^{i-1} d_t \ge 0$. 고로 모든 i에 대해 $s_i > e_i$.

E의 각 정점들은 순방향 점프로 들어와서 역방향 점프로 나가야 하며, S의 각 정점들은 역방향 점프로 들어와서 순방향 점프로 나가야 한다. 그리고 위에서 관찰한 사실에 의해 s_i에서 출발해 e_i에 도착하는 점프는 역방향 점프이다.

따라서, 다음과 같이 방문 순서 p를 구성할 수 있다.

1. $p := (1)$

2. 각 $i = 1, 2, \ldots N-1$에 대해:
   - 어떤 j가 존재해 $s_j = p_i$라면, $p_{i+1} := e_j$
   - 아니라면, $p_{i+1} := \min\left\{x \in \{1, 2, \cdots, n\} \mid x \notin E \land x \notin \bigcup_{j \le i} p_j\right\}$

위 방법으로 구한 방문 순서가 실제로 d_i 조건을 만족함은 쉽게 확인할 수 있다. 시간 복잡도는 $O(N^2)$이다.

부분문제 5

부분문제 4의 풀이에서 $p_{i+1} := \min\left\{x \in \{1, 2, \cdots, n\} \mid x \notin E \land x \notin \bigcup_{j \le i} p_j\right\}$를 계산하는 과정에서 최대 $O(N)$ 시간이 걸려, 총 $O(N^2)$ 시간이 소요되었다.

이를 $\{1, 2, \cdots, n\} \setminus E$을 미리 큐에 넣어 두고 차례로 꺼내는 방식으로 구현하면 전체 $O(N)$ 시간에 동작하게 된다.