

[22.03.11] 긴 사다리 게임

TS사다리

김 태 현



문 제

1~100 선택지를 갖는 사다리 게임이 있다.

세로줄은 모두 그어져 있고 삭제되지 않으며, 가로줄은 없는 상황에서 추가/삭제 된다.

선택지 lineID를 선택하면 (lineID, 0)에서 출발한다.

Y가 증가하는 방향으로 이동하며 가로줄을 만나면 가로줄을 타고 이동한다.

Y좌표가 1,000,000,000이 되면 도착한다.

void init()

모든 세로줄은 그어지고, 가로줄은 없는 상태로 초기화

void add(int mX, int mY)

(mX, mY)와 (mX+1, mY)를 잇는 가로줄 추가

void remove(int mX, int mY)

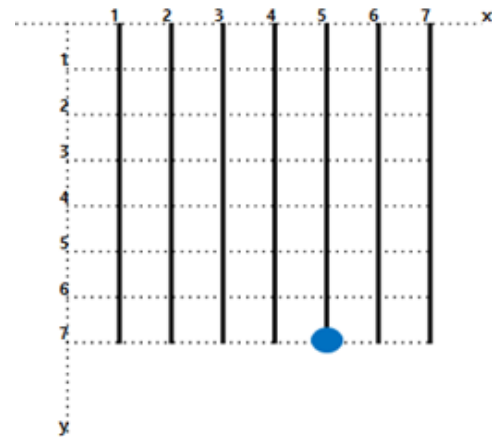
(mX, mY)와 (mX+1, mY)를 잇는 가로줄 삭제

int getCrossCnt(int lineID)

lineID 번을 선택했을 때 지나는 가로줄의 개수 반환
(lineID, 0)에서 출발

int getLineID(int mX, int mY)

(mX, mY)를 지나게 되는 선택지 번호 반환
(mX, mY)는 가로줄과 세로줄이 만나는 지점이 아니며 유일하게 지나는 선택지가 있음을 보장



가로줄이 연속으로 연결된 경우는 존재하지 않음

- add() <= 200,000
- remove() <= 5,000
- getCrossCnt() <= 500
- getLineID() <= 500
- 한 개의 선택 라인이지나는 가로줄 <= 5,000
- mX : 1 ~ 100
- mY : 1 ~ 1,000,000,000

문제 분석

getCrossCnt(), getLineID() : 500회

한 개의 선택 라인이 지나는 가로줄 : 5,000개

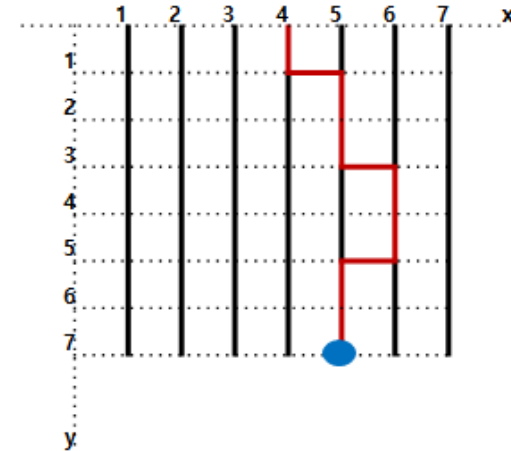
=> 호출시마다 지나는 가로줄 모두 거쳐서 결과 구하는 방식 필요

주요 로직

- 현재 위치(x,y)에서 내려가며 만날 수 있는 가장 가까운 가로줄 검색
- 현재 위치(x,y)에서 올라가며 만날 수 있는 가장 가까운 가로줄 검색
- $(x, y) \sim (x+1, y)$ 가로줄 삭제

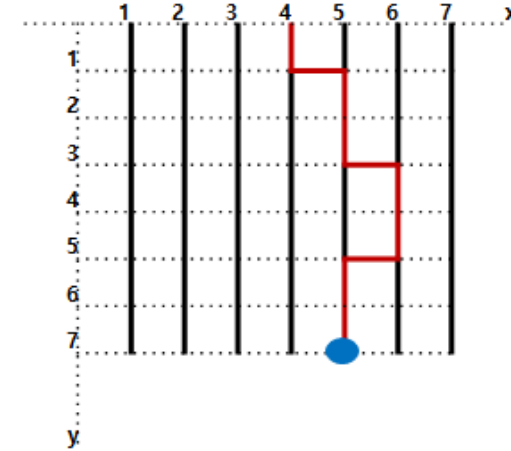
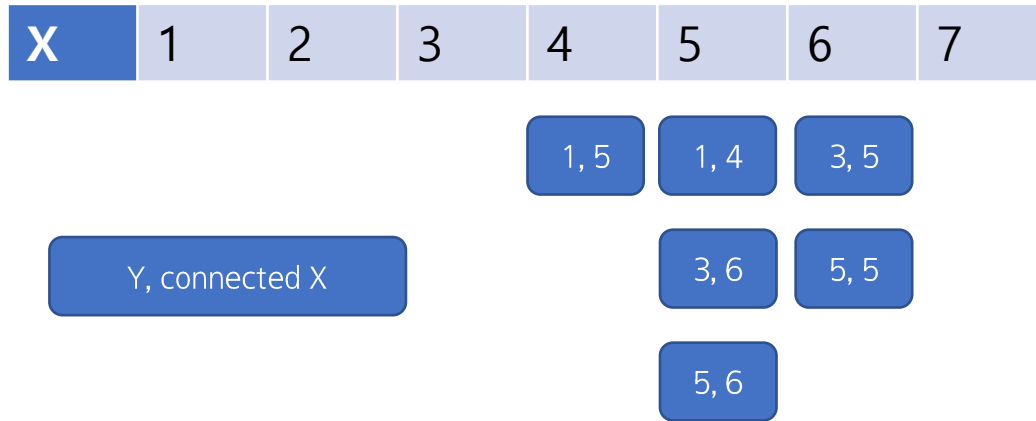
=> line별로 존재하는 가로줄 정보(Y좌표, 연결된 X좌표) 관리

line에 존재할 수 있는 최대 가로줄 : 200,000개



자료 관리

ladder



y 순서 상관 없이 관리 : vector, list

```
add() :  $O(1) * 200,000$   
remove() :  $O(200,000) * 5,000$   
getCrossCnt() :  $O(5,000 * 200,000) * 500$   
getLineID() :  $O(5,000 * 200,000) * 500$ 
```

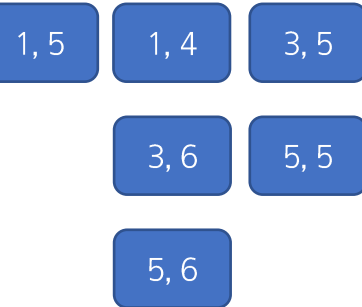
y 오름차순으로 관리 : set/map

```
add() :  $O(2 * \log 200,000) * 200,000$   
remove() :  $O(2 * \log 200,000) * 5,000$   
getCrossCnt() :  $O(5,000 * \log 200,000) * 500$   
getLineID() :  $O(5,000 * \log 200,000) * 500$ 
```

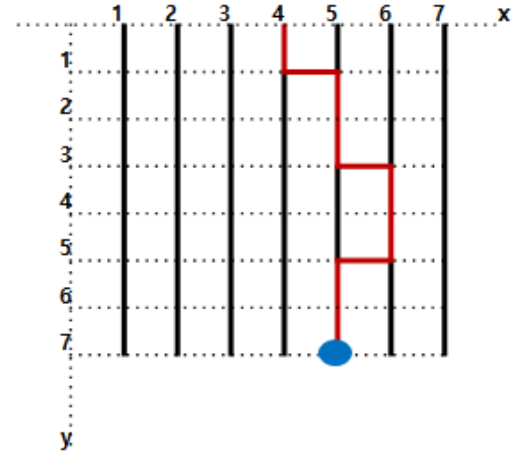
map 활용

ladder

X	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---



key : Y
value : connected X



void add(**int** mX, **int** mY)

```
ladder[mX][mY] = mX+1  
ladder[mX+1][mY] = mX
```

void remove(**int** mX, **int** mY)

```
ladder[mX].erase(mX)  
ladder[mX].erase(mX+1)
```

int getCrossCnt(**int** lineID)

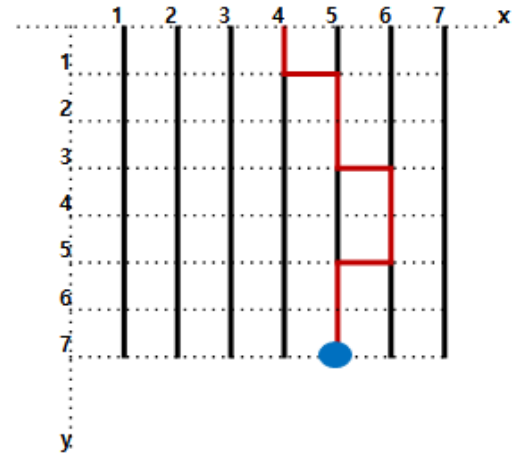
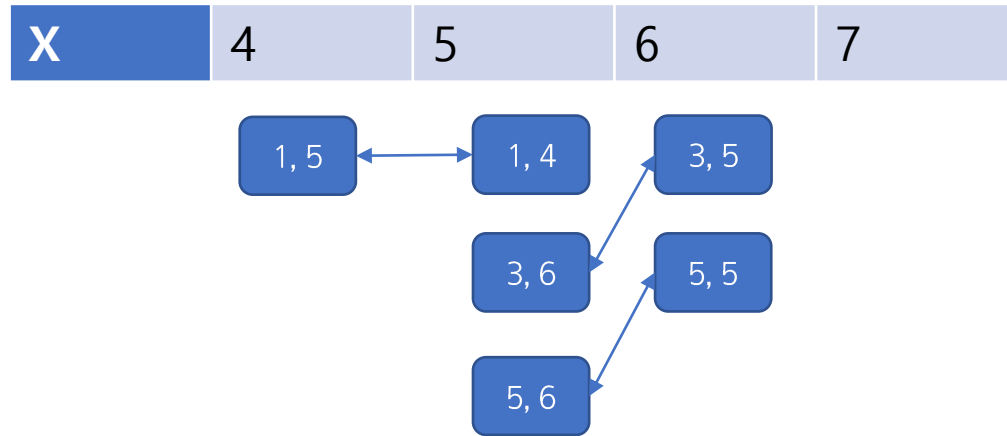
ladder[mX].begin() 가로줄부터 이동
연결된 좌표가 (x,y) 이면 ladder[x].upper_bound(y) 로 다음 가로줄 선택
더 이상 가로줄 없을 때까지 개수 세며 진행

int getLineID(**int** mX, **int** mY)

it = ladder[mX].lower_bound(mY) 로 최초 위치 바로 아래 가로줄 선택
맨 위 가로줄이 아닌동안 반복
--it 하여 가로줄 이동
연결된 좌표가 (x,y) 이면 it = ladder[x].find(y)

개 섰

ladder



서로 연결된 노드 iterator를 기록하여
find() 혹은 upper_bound()로 검색하지 않고 바로 접근한다.

getCrossCnt(), getLineID()

$O(5,000 * \log 200,000) * 500 \Rightarrow O(5,000 * 1) * 500$

감사합니다

