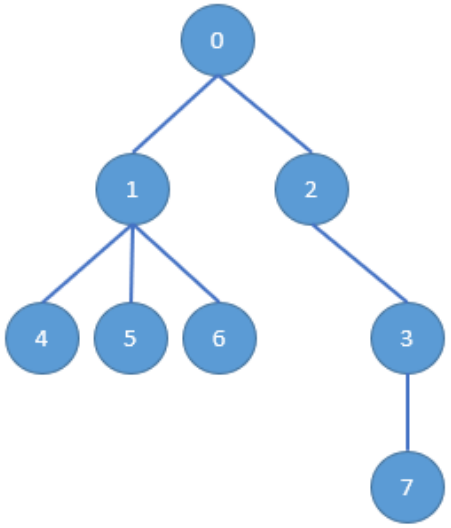
TS트리탐색

트리 구조의 마을



[그림. 1]

경찰은 매시각 목표사건 방향으로 1칸 이동
만약, 목표사건 위치에 있다면 1시간동안 사건 해결
목표사건은 매 시각 새로 설정

목표사건 : 1) 가장 높은 우선순위
2) 가장 이른 시각

1시간 후의 경찰의 위치를 빠르게 파악할 수 있도록 해놓으면,
문제 풀이에 도움이 될 수 있다.

API

1. `void init(int N, int parent[])`

   N개의 마을과 트리 구성 제공

2. `void occur(int timeStamp, int caseID, int townNum, int prior)`

   timeStamp에 caseID 번호의 사건이 townNum에서 prior 우선순위로 발생

3. `void cancel(int timeStamp, int caseID)`

   timeStamp에 caseID 사건 취소

4. `int position(int timeStamp)`

   timeStamp에 경찰의 위치 반환

제약사항

occur() : 100,000회
cancel() : 50,000회
position() : 100,000회

N : 2 ~ 350
townNum : 0 ~ N-1
timeStamp : 0 ~ 5,000,000 (항상 증가)
caseID : 0 ~ 99,999

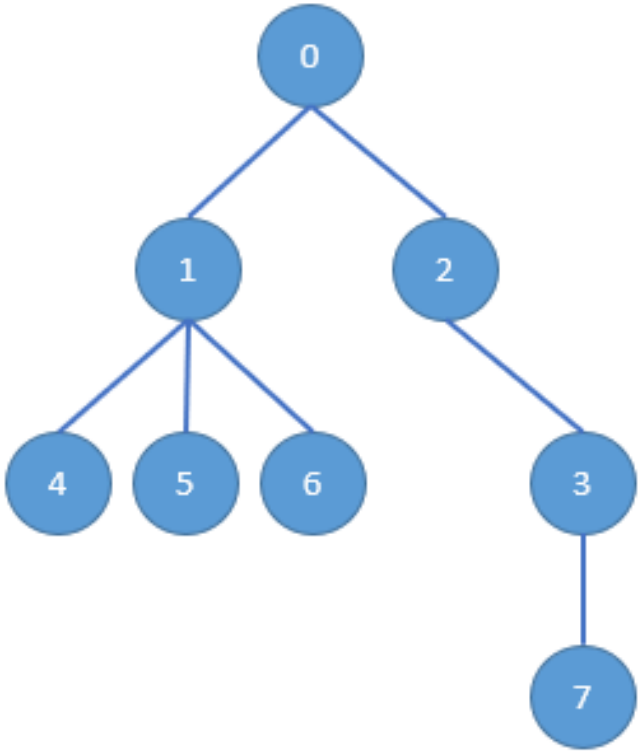
표 생성

for(i : 0 ~ N-1)
for(j : adj[i])
j에서 갈 수 있는 모든 노드 x에 대해 nextTown[i][x] = j
=> dfs, bfs

1시간 후의 경찰의 위치를 빠르게 파악할 수 있도록 해놓으면, 문제 풀이에 도움이 될 수 있다.

1시간 후의 경찰의 위치 (그림. 1)		목표 사건의 발생위치							
		0	*1*	*2*	*3*	*4*	*5*	*6*	*7*
경찰의 위치	*0*	0	1	2	2	1	1	1	2
	1	0	1	0	0	4	5	6	0
	2	0	0	2	3	0	0	0	3
	3	2	2	2	3	2	2	2	7
	4	1	1	1	1	4	1	1	1
	5	1	1	1	1	1	5	1	1
	6	1	1	1	1	1	1	6	1
	7	3	3	3	3	3	3	3	7

[표. 1]



[그림. 1]

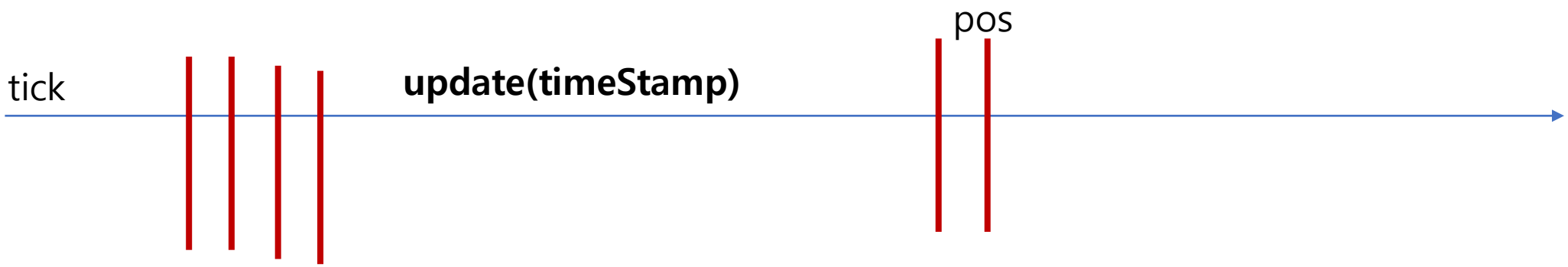
경찰 이동, 사건 관리, 현 시각 목표사건 설정

경찰 정보
1. 현재 시각
2. 경찰 위치

사건 정보
1. 발생 시각
2. 사건 번호
3. 우선순위
4. 발생 마을

목표사건 설정
매번 linear search??? X

우선순위 순으로 관리 : set or pq
1. priority 높은순
2. tick 빠른순



`void occur`
`update(timeStamp)`
사건 추가

`void cancel`
update(timeStamp)
사건 취소

`int position`
update(timeStamp)
return curPos

경찰 이동, 사건 관리, 현 시각 목표사건 설정

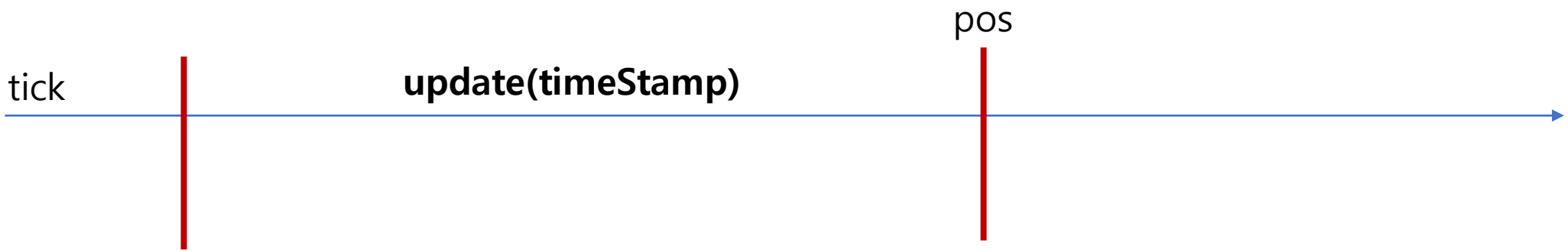
update(timeStamp)

curTick을 1씩 증가하며 진행

1. curTick <= timeStamp
2. 사건이 있는동안 진행
   사건이 없으면 curTick = timeStamp, 종료

매 curTick 마다

1. curTick에서 목표 사건 설정
2. 목표 사건 위치와 현재 위치가 같다면 사건 처리
3. 같지 않다면 nextTown[현재위치][목표사건 위치]로 이동




```
void occur
update(timeStamp)
```
사건 추가

```
void cancel
update(timeStamp)
```
사건 취소

```
int position
update(timeStamp)
return curPos
```