

## 문제정의

사람들이 제각기 일정한 속도로 가고 있을 때,  $t$ 시간이 지나고 나서 존재하는 그룹 수를 구하는 문제(이 때 만약 두 그룹이 만나게 되면 더 느린 속도를 따른다.)

## 문제해결

### 입력

들어오는 입력에 따라 우선순위 큐에 pair를 넣는다. pair의 첫번째 값  $p1$ 은  $p_i + (v_i * t)$ 를 넣고 P2에는  $p_i$ 를 넣는다(우선순위는  $p1$ 을 기준으로 오름차순,  $p1$ 이 같을시  $p2$ 를 기준으로 내림차순이다).

### 그룹간의 포함 관계 찾기

그룹의 포함 관계를 나타내기 위해 스택을 선언한다. 그리고 우선순위 큐에서 pair를 pop하며 큐에서 모든 원소가 나올 때까지 다음을 반복한다.

조건1) 스택이 비어있는 경우

- pop한 pair를 스택에 쌓는다.

조건2) stack의 top에 있는  $p2$ 가 현재 큐에서 꺼낸  $p2$ 보다 작은 경우

- stack의  $p2$ 가 큐의  $p2$ 를 앞질렀다는 뜻이므로  $t$ 시간 내에 만난다는 것이다. 따라서 stack에서 pair를 pop한 뒤 큐의 pair를 스택에 push한다.

조건3) stack의 top에 있는  $p2$ 가 현재 큐에서 꺼낸  $p2$ 보다 큰 경우

- stack의  $p2$ 가 큐의  $p2$ 를 앞지르지 않았으므로 둘은 만나지 않는다. stack에 큐의 pair를 push한다.

반복을 마친 뒤, 스택의 크기를 출력한다.

## 예시

테스트 케이스에 대해서 우선순위 큐와 스택의 관계를 그림으로 표현하면 다음과

Input
5 3
0 1
1 2
2 3
3 2
6 1

stack

stack

stack

stack

stack

stack

stack

stack

stack

stack

stack

priority queue

priority queue

priority queue

priority queue

priority queue

priority queue

priority queue

priority queue

priority queue

priority queue

priority queue

(11,2) (9,3) (9,6) (7,1) (3,0)

(9,3) (9,6) (7,1) (3,0) 조건1

(9,6) (7,1) (3,0) 조건2(2가 3을 앞지름)

(7,1) (3,0) 조건2(3이 6을 포겐다)

조건3(더 이상 앞지르는 경우가 없다)

## 시간복잡도

우선순위큐에서  $n$ 번만큼 push를 하고  $n$ 번만큼 pop하므로 시간복잡도는  $O(n \log n)$ 이다.

## 공간복잡도

모든 그룹이 겹치지 않는 경우 stack에 최대  $n$ 개의 원소가 들어가고 우선순위큐에는 항상  $n$ 개의 원소가 들어가므로 공간복잡도는  $O(n)$ 이 된다.

## 토의

예러케이스를 발견하였다. 4그룹중에 앞에 두그룹과 뒤에 두그룹이 합쳐지고, 만들어진 두 그룹의 속도가 같은 경우 예러가 발생하였다. 이 알고리즘에서는 합해지는 순서가 제대로 정의되지 않은 것 같다