

[22.12.17] 직선왕국

# TS직선왕국

김 태 현



# 문 제

0~N-1 번호를 가지는 N개의 직선 도시 존재  
0번 도시가 수도이며, 인접한 도시를 이동하는데 1시각 소요

## 세금 납부

- A도시에서 세금의 양 만큼의 곡물을 B도시의 창고로 이동

## 관리 파견

- 파견 도착시간 기준으로 가장 곡물이 많을것으로 예상되는 도시 선택
- 같은 경우 ID가 작은 도시 선택
- 도시 선택시 이미 관리가 파견중인 도시는 제외하고
- 대기중인 관리가 없거나 예상 곡물이 0인 경우는 대기

## 파견된 관리 임무 수행

- 파견도시로 이동하여 도착시간에 추가되는 곡물을 포함하여 모든 곡물을 싣고 수도로 즉시 출발
- 수도로 도착하여 곡물을 추가하고 대기상태로 변환

## `void init(int N, int M)`

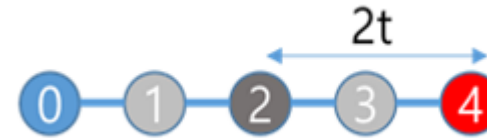
N개의 도시, M명의 관리로 초기화

## `int order(int tStamp, int mCityA, int mCityB, int mTax)`

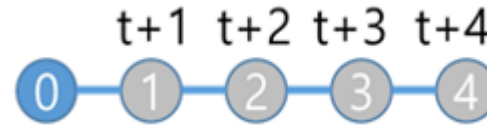
tStamp시각에 mCityA에서 mCityB로 mTax만큼의 곡물 이동  
tStamp까지 모든 상황 처리 후 수도의 곡물 양 반환

## `int check(int tStamp)`

tStamp까지 모든 상황 처리 후 수도의 곡물 양 반환



도시 간 이동은 번호 차이만큼의 시간이 소요된다.



t시각에 관리 파견 시, 예상 곡물을 판별하는 시각

## [제약사항]

N : 도시 개수  $\leq 200$

M : 관리 수  $\leq 50$

tStamp : 1 ~ 100,000,000

order()  $\leq 10,000$

check()  $\leq 10,000$

# TimeFlow 유형

함수 호출마다 tStamp가 주어진다.  
함수가 호출되는 tStamp 사이에도 상황은 진행된다.

매 시각에 발생할 수 있는 모든 이벤트를 생각해보자.  
이벤트의 우선순위와 서로의 연관관계를 고려한다.

함수 호출시마다 `update(tStamp)` 를 진행한다.

# 이벤트 종류

- |           |            |                    |
|-----------|------------|--------------------|
| 1. ADD    | 곡물 납부 완료   | (tStamp, cid, cnt) |
| 2. EXPECT | 예상 곡물 추가   | (tStamp, cid, cnt) |
| 3. ARRIVE | 파견 도시 도착   | (tStamp, cid)      |
| 4. BACK   | 파견 복귀      | (tStamp, cid, cnt) |
| 5. SEND   | 파견 판별 & 출발 | (tStamp)           |

도시 별 창고의 곡물량, 예상 곡물량, 파견 상태 관리

[illegible]

# 관리 파견

## 파견 할 수 있는 상태

대기중인 관리가 있고, 예상 곡물량이 있는 파견중이지 않은 도시가 있는 경우

## 파견을 고려해야 하는 경우

1. 파견관리가 복귀한 경우                      BACK
2. 예상 곡물량이 추가되는 경우                  EXPECT

=> 특정 시각의 BACK, EXPECT 를 모두 수행한 이후 SEND 수행

## 파견 도시 선정

### 1. linear search

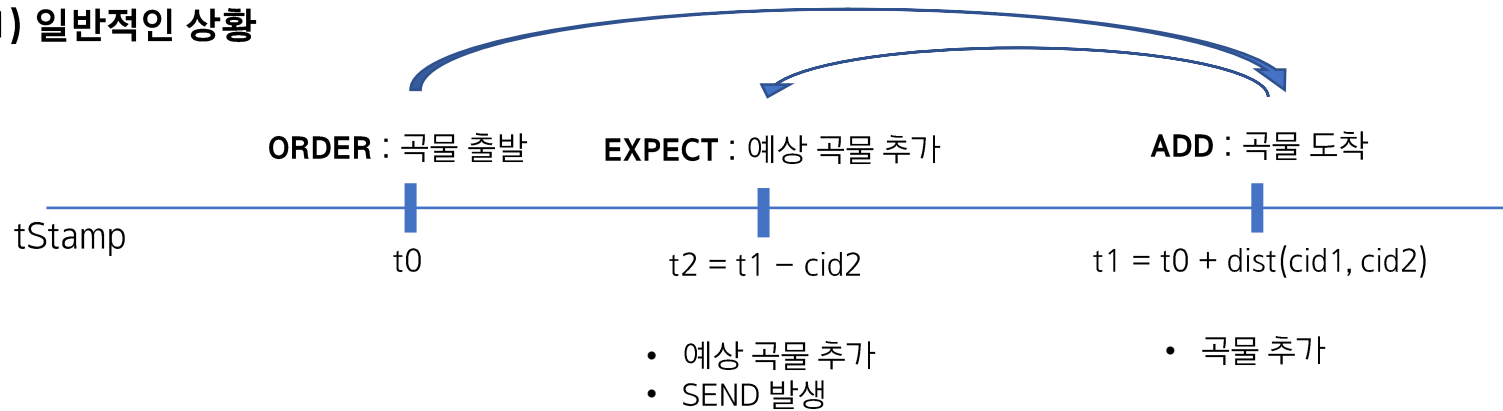
- 매번 모든 도시를 확인
- 관리가 파견되지 않았으면서 우선순위 높은 도시 선정
- 선정된 도시는 파견 불가능 상태로 변경, 관리가 복귀하면 파견 가능 상태로 변경

### 2. set

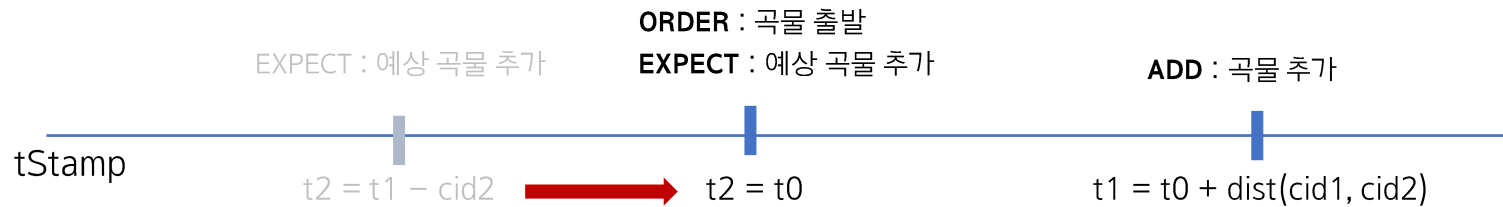
- 파견 가능한 도시의 { 예상 곡물량, 도시 번호 }를 우선순위 순으로 관리
- 파견 중이지 않은 도시이면서 예상 곡물이 존재하는 경우에만 set에 등록
- BACK, EXPECT 수행 시, 업데이트

# ADD, EXPECT

## 1) 일반적인 상황



## 2) 예상 곡물 추가시간이 출발시간 이전인 경우



만약,  $t_2$ 가  $t_0$ 보다 작으면  
 $t_0$ 시각에 바로 예상곡물이 추가된다.

# SEND, ARRIVE, BACK



## 이벤트 우선순위

1. tStamp 빠른 순
2. EXPECT , BACK > SEND  
ADD > ARRIVE

감사합니다

