

[22.10.08] 일정관리4

TS일정관리4

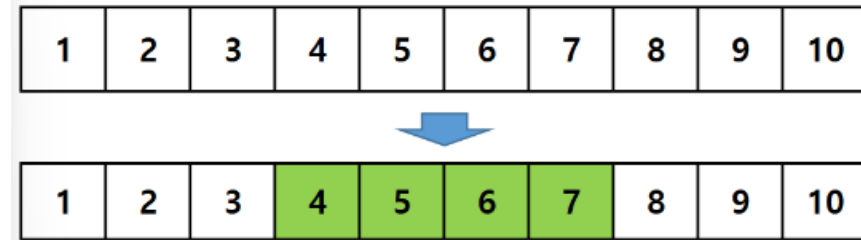
김 태 현



문 제

제목, 시작일, 종료일을 가지는 일정들이 존재하며 아래 네가지 명령을 수행한다

- ① 일정 등록
 - a. 일반모드 : 겹치는 일정이 있으면 등록 실패
 - b. 강제모드 : 겹치는 일정 모두 삭제 후 일정 등록
- ② 일정 찾기 : 특정 일자를 포함한 일정 찾기
- ③ 일정 삭제
- ④ 가장 긴 빈 일정 찾기 : 어떤 일정도 등록되지 않은 가장 빈 일정 찾기



[Fig. 2]

start=4, end=7 인 일정 등록 예시

void init(**int** N)

달력의 범위가 1 ~ N 으로 존재
 $10 \leq N \leq 1,000,000$

int addSchedule(**char** title[], **int** start, **int** end, **int** forced) $\leq 30,000$

제목(title), 시작일(start), 종료일(end)인 일정 등록 등록 성공시 1, 실패시 0 반환
 forced = 0 : 일반모드, 1 : 강제모드
 title : 소문자, 길이 1~10, 유일하며 이전에 요청된적 없음
 start : 1 ~ N end : start ~ N

RESULT getSchedule(**int** mDay)
 $\leq 20,000$

mDay 일정에 등록된 일정의 정보(title, start, end)
 를 RESULT에 등록 후 반환

int deleteSchedule(**char** title[])
 $\leq 5,000$ 회

title 일정 삭제
 이미 삭제됐거나 등록되지 않은 경우 실패
 성공시 1, 실패시 0 반환

int findEmptySchedule() ≤ 100

가장 긴 빈 일정 길이 반환

문제 파악

※ 일정 개수 = 30,000 , 달력 크기 = 1,000,000

1. addSchedule() <= 30,000 , getSchedule() <= 20,000

구간에 겹치는 일정 있는지 파악

겹치는 모든 일정 검색

=> linear search 시, $O(\text{일정 개수} * \text{호출 횟수})$: 최적화 필수

2. deleteSchedule() <= 5,000

제목으로 일정 존재 여부 확인과 일정 삭제

=> linear search 시, $O(\text{일정 개수} * \text{호출 횟수} * \text{이름 비교 비용})$: 최적화 권장

3. findEmptySchedule() <= 100

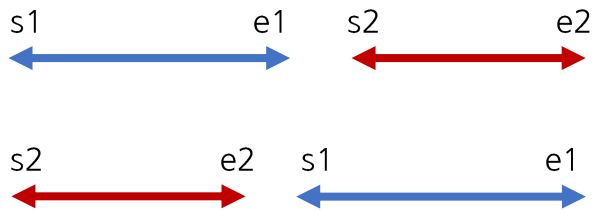
모든 빈 구간 중 길이가 가장 긴 구간의 길이 파악

=> linear search 시, $O(\text{일정 개수} * \text{호출 횟수})$: 최적화 필요X

일정 겹치는지 판별

안겹치는 경우

$e1 < s2 \parallel e2 < s1$



겹치는 경우

$\neg(e1 < s2 \parallel e2 < s1)$



$e1 >= s2 \ \&\& \ e2 >= s1$

addSchedule(), getSchedule()

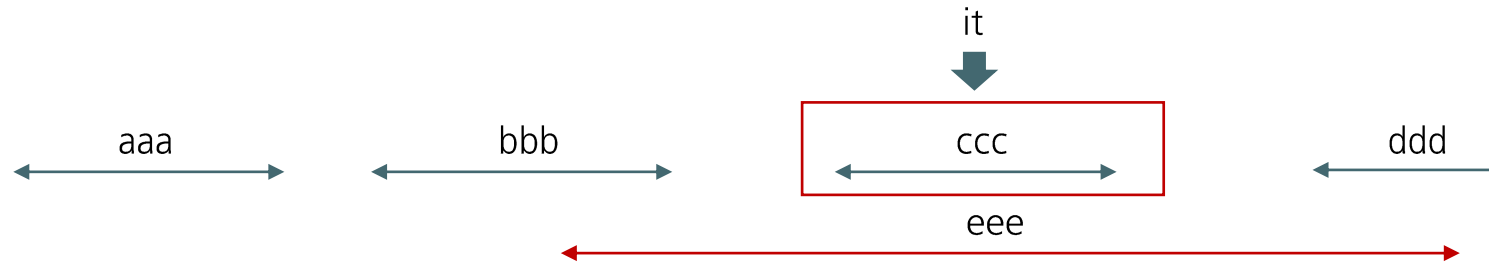
set, map lower_bound 활용

1. key = {start, end}, value = title

구간 {s,e} 검색시, $it = \text{lower_bound}(\{s, 0\})$: s 이후(s 포함)에 시작하는 가장 빠른 구간 선택

1) s 이전에 시작하면서 겹칠 수 있는 일정 : $it-1$ 유일

2) s 이후에 시작하면서 겹칠 수 있는 일정 : it 부터 겹치지 않는 일정 나올때까지 연속



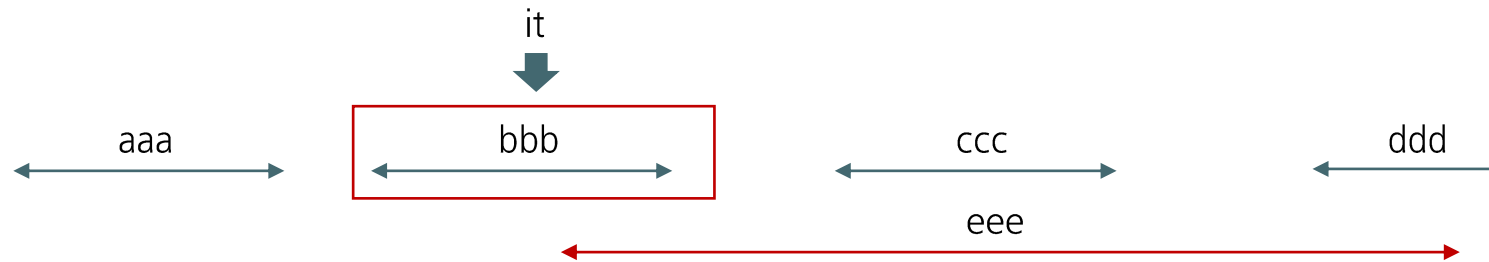
addSchedule(), getSchedule()

set/map lower_bound 활용

2. key = {end, start}, value = title

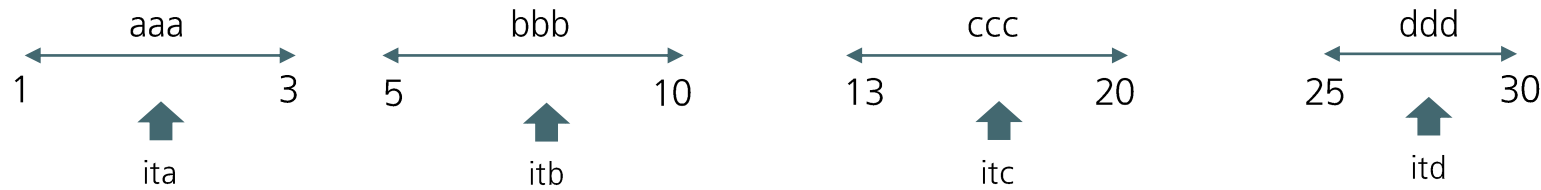
구간 {s,e} 검색시, $it = \text{lower_bound}(\{s, 0\})$: s 이후(s 포함)에 끝나는 가장 빠른 구간 선택

s 이후에 끝나면서 겹칠 수 있는 일정 : it 부터 겹치지 않는 일정 나올때까지 연속



deleteSchedule()

이름에 따른 일정 삭제 : hash 활용(unodered_map)



1) **key** = title
value = {start, end} or {end, start}

title	{start, end}
aaa	{1, 3}
bbb	{5, 10}
ccc	{13, 20}
ddd	{25, 30}

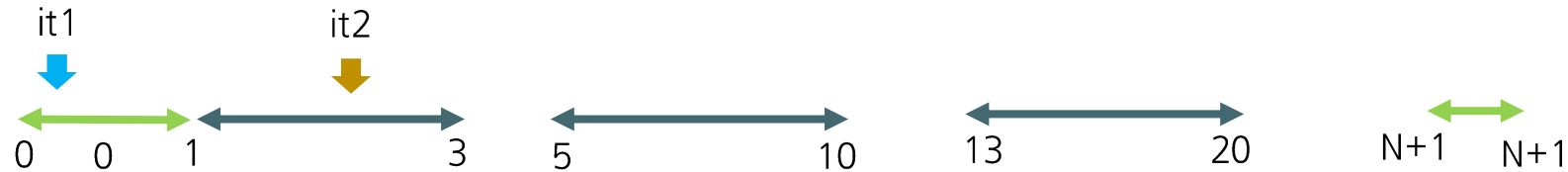
2) **key** = title
value = set/map에 등록된 iterator

title	{start, end}
aaa	ita
bbb	itb
ccc	itc
ddd	itd

findEmptySchedule()

set/map 노드 linear search

- 1) $\{0,0\}, \{N+1, N+1\}$ 노드 등록하고 $it1 = \text{begin}()$, $it2 = \text{begin}() + 1$ 에서부터 1씩 증가하며
두 일정간 빈 공간 크기의 최대값 반환
빈공간 크기 = $it2.start - it1.end - 1$



- 2) $end = 0$, $it = \text{begin}()$ 으로 하여 it 와 end 사이의 빈 공간의 최대값 반환
 end = 이전 일정의 종료일이고, $it+1$ 하기전에 it 의 end 로 업데이트
빈공간 크기 = $it.start - end - 1$



감사합니다

