



[22.07.15] 로그인대기열

# TS로그인대기열

김 태 현

# 문 제

1. 많은 사람이 접속하여 이용하는 사이트가 있다.  
접속 허용인원을 초과하면 로그인 대기열에 순차적으로 추가된다.  
문제에서는 항상 접속 허용인원이 차 있는 상태로 주어진다.
2. 아이디를 대기열에 추가, 삭제하는 함수  
접속시키는 함수  
대기 번호를 확인하는 함수를 구현한다.

---

## 1) void init()

접속 인원은 모두 차 있고, 로그인 대기열은 비어 있는 상태로 초기화

## 2) void loginID(char strID[10])

strID(5~9자리 소문자)를 로그인 대기열에 추가  
이미 대기열에 있는 경우, 삭제 후 새롭게 추가

## 3) int closeIDs(char prefix[10])

prefix(3~5자리 소문자)로 시작하는 모든 아이디를 대기열에서 삭제  
삭제된 아이디 개수 반환

## 4) void connectCnt(int accessCnt)

대기열에 추가된 순서대로 accessCnt개가 접속되어 대기열에서 삭제  
대기열에 있는 아이디 개수가 accessCnt 이상임을 보장

## 5) int waitOrder(char strID[10])

strID가 대기열에 추가된 순서대로 몇 번째에 대기중인지 순서 반환  
대기열에 strID가 없는 경우 0 반환  
strID가 loginID() 에서 접속한 적이 없을 수도 있음

## ※ 제약사항

loginID() <= 50,000  
closeIDs() <= 5,000 , 삭제되는 아이디 최대 개수 20  
connectCnt(), waitOrder() <= 각 1,000

# Naive

로그인 대기열 최대 원소 개수  $N \leq 50,000$

index	0	1	2	3	4	5	6	7
strID	faster	first	second	third	firster	fast	fasti	

array



**void loginID(char strID[10])**  $\leq 50,000$

대기열 검색해서 strID 있으면 삭제  
마지막에 등록

**void connectCnt(int accessCnt)**  $\leq 1,000$

앞에서부터 accessCnt개 삭제

**int closeIDs(char prefix[10])**  $\leq 5,000$

대기열에 prefix 로 시작하는 ID  
전부 삭제

**int waitOrder(char strID[10])**  $\leq 1,000$

strID 검색해서 index+1 출력

모든 함수  $O(N * a)$

a는 문자열 비교 연산 있는 경우

# Naïve2

hash 추가, head 추가

htab:unordered\_set

key=strID

로그인 대기열 최대 원소 개수  $N \leq 50,000$

array

index	0	1	2	3	4	5	6	7
strID	faster	first	second	third	firster	fast	fasti	

↑  
head

↑  
tail

**void loginID(char strID[10])**  $\leq 50,000$

htab에 있는 경우, 해당 index 삭제  
마지막에 등록 후, htab[strID]

**void connectCnt(int accessCnt)**  $\leq 1,000$

head += accessCnt

**int closeIDs(char prefix[10])**  $\leq 5,000$

대기열에 prefix 로 시작하는 ID  
전부 삭제

**int waitOrder(char strID[10])**  $\leq 1,000$

htab에 있는 경우만  
strID 검색해서 index+1 출력

# Naïve3

hash에 index포함, 실제로 삭제하지 않음, valid[] 추가

htab:unordered\_map

key=strID  
value=index

로그인 대기열 최대 원소 개수  $N \leq 50,000$

array

index	0	1	2	3	4	5	6	7
strID	faster	first	second	third	firster	fast	fasti	
valid	1	0	1	1	1	1	1	

  
head

  
tail

**void loginID(char strID[10])  $\leq 50,000$**

대기열에 새롭게 추가하고 htab[strID] = 추가된 index  
valid[추가된 index] = 1  
기존에 존재했으면 valid[기존 index] = 0

**void connectCnt(int accessCnt)  $\leq 1,000$**

valid 1인 원소를 accessCnt개 될때까지 head++

**int closeIDs(char prefix[10])  $\leq 5,000$**

대기열에서 prefix 로 시작하는 index 전부 valid 0으로

**int waitOrder(char strID[10])  $\leq 1,000$**

htab에 있는 경우만 head부터 검색하여 찾으며  
valid 1 개수 반환

# closeIDs 개선 최종

htab:unordered\_map

key=strID  
value=index

로그인 대기열 최대 원소 개수 N <= 50,000

array

index	0	1	2	3	4	5	6	7
strID	faster	first	second	third	firster	fast	fasti	
valid	1	0	1	1	1	1	1	

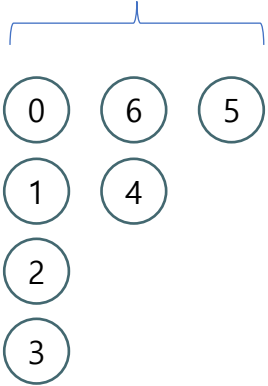
↑  
head

↑  
tail

V[hash]:vector

평균 : 50,000 / 17,576 = 3.. 개  
문제 조건에 의해 최대 20개

앞 세자리 hash
fas
fir
sec
thi
..



26³ = 17,576

void loginID(char strID[10]) <= 50,000

대기열에 새롭게 추가하고 htab[strID] = 추가된 index  
앞세자리로 hash값 생성하여 V[hash]에 index 추가  
valid[추가된 index] = 1  
기존에 존재했으면 valid[기존 index] = 0

int closeIDs(char prefix[10]) <= 5,000

prefix 앞 세자리로 hash값 생성하여 v[hash] 전부 검색

- valid[x]가 0 이면 삭제
- prefix 만족하면 삭제, htab 삭제, valid[x] = 0

void connectCnt(int accessCnt) <= 1,000

valid 1인 원소가 accessCnt개 될때까지 head++  
valid 0으로 변경, htab 삭제

int waitOrder(char strID[10]) <= 1,000

htab에 있는 경우만 head부터 검색하여 찾으며  
valid 1 개수 반환

\* hash = 앞 세자리 26진법(a:0, z:25)

감사합니다

