

[22.02.12] 순서 맞추기 게임

TS랭크 맞추기

김 태 현



문 제

- 0~999 범위의 id값에 0~999의 서로 다른 rank 존재
- 아래 두가지 질문을 통해 모든 id의 rank를 구해야 한다.
 1. 5개 이상의 id 중 가장 낮은 rank
 2. 5개 이상의 id 중 가장 높은 rank
- 1,200회 이하의 질문으로 모든 rank를 구해야 한다.
(실제 검정에선 모든 TC의 질문 총합이 일정 값 이하인 경우에 합격)

int query(int M, int sub[], int opt)

main.cpp에서 구현되어 있는 함수

sub[0:M-1] 에 저장된 id들의 가장 높은/낮은 rank 반환

opt=0 : 가장 낮은 rank

opt=1 : 가장 높은 rank

void getRank(int retRank[N])

user.cpp에서 구현해야 하는 함수

test case당 10회 호출

query 함수 호출을 통해 모든 rank를 retRank에 담는다.

retRank[id] = id의 rank

일반적인 접근

Main

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

User

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank															

- 5개 선택해서 큰 rank 구하기
- 1개 추가해서 큰 rank 구하기
 1. 결과가 다른 경우 : 추가한 id의 rank 확정
 2. 결과가 같은 경우 : 기존 5개를 1개씩 빼고 큰 rank 구해보면서 1개 확정
- 확정된 1개 제외하고 반복
- 한 개를 확정하기 위해 1~4번의 쿼리 진행

일반적인 접근

Main

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

User

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank															

query(5, [0,1,2,3,4], 1) = 9

- 5개 선택해서 큰 rank 구하기
- 1개 추가해서 큰 rank 구하기
 1. 결과가 다른 경우 : 추가한 id의 rank 확정
 2. 결과가 같은 경우 : 기존 5개를 1개씩 빼고 큰 rank 구해보면서 1개 확정
- 확정된 1개 제외하고 반복
- 한 개를 확정하기 위해 1~4번의 쿼리 진행

일반적인 접근

Main

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

User

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank															

$\text{query}(6, [0,1,2,3,4,5], 1) = 9$

- 5개 선택해서 큰 rank 구하기
- 1개 추가해서 큰 rank 구하기
 1. 결과가 다른 경우 : 추가한 id의 rank 확정
 2. 결과가 같은 경우 : 기존 5개를 1개씩 빼고 큰 rank 구해보면서 1개 확정
- 확정된 1개 제외하고 반복
- 한 개를 확정하기 위해 1~4번의 쿼리 진행

일반적인 접근

Main

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

User

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank															

query(5, [1,2,3,4,5], 1) = 8

- 5개 선택해서 큰 rank 구하기
- 1개 추가해서 큰 rank 구하기
 1. 결과가 다른 경우 : 추가한 id의 rank 확정
 2. 결과가 같은 경우 : 기존 5개를 1개씩 빼고 큰 rank 구해보면서 1개 확정
- 확정된 1개 제외하고 반복
- 한 개를 확정하기 위해 1~4번의 쿼리 진행

일반적인 접근

Main

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

User

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9														

$\text{query}(5, [1,2,3,4,5], 1) = 8$

- 5개 선택해서 큰 rank 구하기
- 1개 추가해서 큰 rank 구하기
 1. 결과가 다른 경우 : 추가한 id의 rank 확정
 2. 결과가 같은 경우 : 기존 5개를 1개씩 빼고 큰 rank 구해보면서 1개 확정
- 확정된 1개 제외하고 반복
- 한 개를 확정하기 위해 1~4번의 쿼리 진행

일반적인 접근

Main

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

User

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9														

$\text{query}(6, [1,2,3,4,5,6], 1) = 8$

- 5개 선택해서 큰 rank 구하기
- 1개 추가해서 큰 rank 구하기
 1. 결과가 다른 경우 : 추가한 id의 rank 확정
 2. 결과가 같은 경우 : 기존 5개를 1개씩 빼고 큰 rank 구해보면서 1개 확정
- **확정된 1개 제외하고 반복**
- 한 개를 확정하기 위해 1~4번의 쿼리 진행

일반적인 접근

Main

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

User

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9		8												

$\text{query}(5, [2,3,4,5,6], 1) = 8$

$\text{query}(5, [1,3,4,5,6,7], 1) = 7$

- 5개 선택해서 큰 rank 구하기
- 1개 추가해서 큰 rank 구하기
 1. 결과가 다른 경우 : 추가한 id의 rank 확정
 2. 결과가 같은 경우 : 기존 5개를 1개씩 빼고 큰 rank 구해보면서 1개 확정
- 확정된 1개 제외하고 반복
- 한 개를 확정하기 위해 1~4번의 쿼리 진행

일반적인 접근

Main

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

User

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9		8					12							

query(6, [1,3,4,5,6,7], 1) = 12

- 5개 선택해서 큰 rank 구하기
- 1개 추가해서 큰 rank 구하기
 1. 결과가 다른 경우 : 추가한 id의 rank 확정
 2. 결과가 같은 경우 : 기존 5개를 1개씩 빼고 큰 rank 구해보면서 1개 확정
- 확정된 1개 제외하고 반복
- 한 개를 확정하기 위해 1~4번의 쿼리 진행

일반적인 접근

Main

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

User

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9		8					12	13						

query(6, [1,3,4,5,6,8], 1) = 13

- 5개 선택해서 큰 rank 구하기
- 1개 추가해서 큰 rank 구하기
 1. 결과가 다른 경우 : 추가한 id의 rank 확정
 2. 결과가 같은 경우 : 기존 5개를 1개씩 빼고 큰 rank 구해보면서 1개 확정
- 확정된 1개 제외하고 반복
- 한 개를 확정하기 위해 1~4번의 쿼리 진행

일반적인 접근

Main

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

User

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9		8			7		12	13	11	10	14	5	6	

최종적으로 rank가 가장 낮은 5개가 남음

- $\text{query}(5, [\text{남은 id 한 개} + \text{채워진 4개}], 0) = \text{남은 id 한 개의 rank}$
- 한번의 query로 한 개씩 구할 수 있음

※ 가장 높은(낮은) rank의 id 4개를 알면 query 한번에 한 개씩 확정 가능

ex) $\text{query}(5, [1, 7, 8, 9, 10], 0) = 0$

개 선택

가장 낮은 rank인 id 4개를 구한 뒤, 나머지 996개는 한 번으로 구하기

총 쿼리 수 = 4개 구하는 쿼리 수 + 996

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

4개 구하는 방법_1

- 5개씩 확인하며 rank 0,1,2,3이 있는 구간을 찾는다. : 1,000/5 = 200회
[0:4] , [5:9] , [10:14] , ... , [995:999]
- 찾은 구간에서 각각의 값을 찾는다. : 각각 최대 4회

query 수 : 200 + 4*4 미만




조금 아쉽다..

추가 개선

가장 낮은 rank인 id 4개를 구한 뒤, 나머지 996개는 한 번으로 구하기

총 쿼리 수 = 4개 구하는 쿼리 수 + 996

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

 s
  mid
  e

4개 구하는 방법_2

- I. rank 0부터 찾는데 전체 구간에서 절반씩 확인하며 구간을 좁혀나간다. : $\log_2 1000$ 회
 $mid = (s+e)/2$ 만약, $[s:mid]$ 가 5개 미만이면 $mid=s+4$
 $query(mid-s+1, [s:mid], 0) == x : e = mid$
 $query(mid-s+1, [s:mid], 0) != x : s = mid+1$
- II. $[s:e]$ 가 5개 이하면 이 구간에서 값을 찾는다. : 최대 4회


쿼리수=14*4 미만

추가 개선

가장 낮은 rank인 id 4개를 구한 뒤, 나머지 996개는 한 번으로 구하기

총 쿼리 수 = 4개 구하는 쿼리 수 + 996

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2


s mid e

4개 구하는 방법_2

- I. rank 0부터 찾는데 전체 구간에서 절반씩 확인하며 구간을 좁혀나간다. : $\log_2 1000$ 회
mid = (s+e)/2 만약, [s:mid]가 5개 미만이면 mid=s+4
query(mid-s+1, [s:mid], 0) == 0 : e = mid
query(mid-s+1, [s:mid], 0) != 0 : s = mid+1
- II. [s:e] 가 5개 이하면 이 구간에서 값을 찾는다. : 최대 4회



쿼리수=14*4 미만

추가 개선

가장 낮은 rank인 id 4개를 구한 뒤, 나머지 996개는 한 번으로 구하기

총 쿼리 수 = 4개 구하는 쿼리 수 + 996

id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
rank	9	0	8	1	3	7	4	12	13	11	10	14	5	6	2

 s  e

4개 구하는 방법_2

- I. rank 0부터 찾는데 전체 구간에서 절반씩 확인하며 구간을 좁혀나간다. : $\log_2 1000$ 회
mid = (s+e)/2 만약, [s:mid]가 5개 미만이면 mid=s+4
query(mid-s+1, [s:mid], 0) == 0 : e = mid
query(mid-s+1, [s:mid], 0) != 0 : s = mid+1
- II. [s:e]가 5개 이하이면 이 구간에서 값을 찾는다. : 최대 4회

쿼리수 = 14 * 4 미만

감사합니다

