S221 - Hotel Rankings

Fuzen Ng {yfng}

2022-01-29

香港電腦奧林匹克競賽
Hong Kong Olympiad in Informatics

Background

Problem idea by Fuzen Ng

Preparation by Fuzen Ng, Bryan Chung

Figures by Christy Cheng

Problem Restatement

There are *N* hotels

Given *M* ratings, each gives an integer score of 1 to *K* to one of the *N* hotels

Define R_i = Rank of hotel i = 1 + number of hotels with total score larger than hotel i's score

Given a target ranking *R'*

Find the minimum number of new ratings needed to attain the ranking *R'*

Sample Input	Sample output
4 5 5 1 2 4 2 3 3 1 4 4 2 1 5 4 1	2
2 2 5 1 2 1 3 2 5	1

Statistics

Task	Attempts	Max	Mean	Std Dev
S221 - Hotel Rankings	75	100	51.52	41.082

Subtasks						
6: 67	7: 60	12: 49	17: 40	20: 31	13: 33	25: 29

First solved by **dbselisonlee** at **0:12:10**

香港電腦奧林匹克競賽
Hong Kong Olympiad in Informatics

SUBTASKS

For all cases:
$1 \le N \le 2 \times 10^5$
$0 \le M \le 2 \times 10^5$
$1 \le K \le 10^9$
$1 \le R'_i \le N$ for $1 \le i \le N$
$1 \le H_i \le N$ for $1 \le i \le M$
$1 \le S_i \le K$ for $1 \le i \le M$

	Points	Constraints
1	6	$N = M = 2$ $R'_1 = 1, R'_2 = 2$
2	7	$N = 2, K = 1$
3	12	$1 \le N, M, K \le 5000$ $R'_i = i$ for $1 \le i \le N$ $M = N, H_i = i$ for $1 \le i \le M$
4	17	$R'_i = i$ for $1 \le i \le N$
5	20	$1 \le N, M, K \le 5000$
6	13	$M = 0$
7	25	No additional constraints

Subtask 1

Subtask 1 (6%): $N = M = 2$, $R'_1 = 1$, $R'_2 = 2$

- Two ratings for two hotels
- $R'_1 = 1$
- Add new ratings for hotel 1 until its score is larger than hotel 2

- Or if-then-else
  - answer must be 0 to 3

Sample Input 2	Sample output 2
2 2 5 1 2 1 3 2 5	1

Subtask 2

Subtask 2 (7%): $N = 2$, $K = 1$

- Two hotels
- All ratings has a score of 1
- Count the number of ratings towards the two hotels
- If the ranking is matched
  - answer = 0
- Else if the two hotels has the same ranking ($R'_1 = R'_2 = 1$),
  - answer = difference between the score of the hotels
- Else
  - answer = difference between the score of the hotels + 1

Subtask 3

Subtask 3 (12%): $1 \le N, M, K \le 5000$, $M = N$,
$R'_i = i$ *for* $1 \le i \le N$,
$H_i = i$ *for* $1 \le i \le M$

- $R'_i = i$
  - Hotel 1 must have the highest score
  - Hotel 2 must have the second highest score
  - ...
- Greedily add ratings from hotel N-1 to hotel 1
- Add minimum number of ratings and scores so that score of hotel i > score of hotel i+1
  - Add ratings with score K one by one
  - Change the last added score so that score of hotel i = (score of hotel i+1) + 1

Subtask 4

Subtask 4 (17%): $R'_i = i$ *for* $1 \le i \le N$

- $R'_i = i$
  - Hotel 1 must have the highest score
  - Hotel 2 must have the second highest score
  - ...
- Greedily add ratings from hotel N-1 to hotel 1
- Add minimum number of ratings and scores so that score of hotel i > score of hotel i+1
  - Target score of hotel i = (score of hotel i+1) + 1
  - Number of ratings added = (target - score - 1) / K + 1
  - O(1) for each hotel
- Time complexity: O(N)

Subtask 5

Subtask 5 (20%): $1 \le N, M, K \le 5000$

- Any approaches with time complexity = $O(N^2)$
  - Loops through the N hotels and looks for an unvisited hotel with the lowest ranking
    - Calculate the number of ratings needed in O(1)
    - Mark the hotel as visited
    - Repeat *N* times
  - Sort the hotels by their rankings
    - Add new ratings to the hotels one by one
- Handle hotels with the same rankings carefully
  - They must have the same score
  - Find the highest score among themselves
  - Compare the highest score with the previous score
    - Target score = max(highest score, previous score + 1)
  - each hotel needs to add a score of (target score - score of hotel)

Subtask 6

Subtask 6 (13%): $M = 0$

- Sort the rankings
- Set the scores of the hotels greedily such that the number of ratings needed is minimized
  - Score of the hotel(s) with the lowest ranking must be 0
  - Score of the hotel(s) with the second lowest ranking must be 1
  - ...
- For each hotel, calculate the number of ratings needed for its score
  - (Score - 1) / K + 1

Full Solution

Subtask 7 (25%): No additional constraints

- Combine the two possible O(N^2) solutions mentioned
- Sort the hotels by their rankings
- Start from the lowest ranking hotel(s)
- Handle hotels with the same rankings as mentioned
- Calculate the number of ratings needed for each hotel in O(1)

- Time complexity: O(N log N)

Sorting with Self-defined Conditions

- In this task, it is more convenient to sort the hotels by their rankings, then by their scores if their ranking is the same
- In c++, you may define boolean operators in struct besides using pairs

```
struct Hotel {
    int rank, hotel;
    long long sum;
    bool operator < (const Hotel &T) const {
        if (rank != T.rank) return rank < T.rank;
        return sum < T.sum;
    }
};
```