

Challenge 20 - Tuenti star

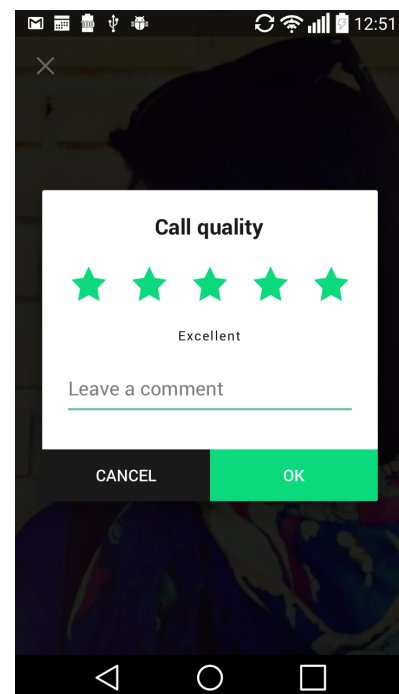
[« Prev](#) [Next »](#)

NOTE: This is an optimization problem where the process of submitting your solution is slightly different. Your target is to get the best solution possible, and you may try as many times as you want. When you submit a solution, it will be validated. If the solution is invalid, you will get an error message. If the solution is valid, you will get a score. You can submit as many times as you want, and only the best score will be stored.

Our main goal at Tuenti is to make our product great so that our customers are satisfied. An important part of our product is our VozDigital offering. With VozDigital, you can call mobiles and landlines from both mobile data or WiFi networks. When a VozDigital call is made, a connection is made to our servers, which transform and redirect it through traditional PSTN. At the end of the call, we offer our customers a dialog to rate the quality of the call which helps us to obtain some metrics about the quality of our service.

There are many different parameters that affect the quality of a call, but we'll only consider the ones which we consider to be most important:

- The ability to successfully make a call or not. Customers will be more satisfied if they can actually make calls :).
- The establishment time of a call. Customers may tolerate some delay until their call is established, but the less the better.
- The delay during a call. The shorter the lag,



the better.

To support the calls our customers make, we have different Points of Presence (*POP*) distributed strategically. These POPs have a limited capacity and they will deny a call if this limit is surpassed. Fortunately, calls can be queued even if the limit is reached, and established later when an existing call terminates. So, your task is easy. You just need to route and enqueue our customer calls properly to make the best use of our POPs. We need your help to make our customers as happy as possible!

We have the relevant information about POPs: their position (represented as a point in space with two coordinates X and Y) and their capacity C (the number of concurrent calls it supports). Once the limit is reached, they have an infinite queue for pending calls. Once a call is established, it cannot be removed until it finishes.

Fortunately, we also have a pretty good profile of our customers, so we know not only when the users will make a call and its duration but how they will rate each call. The rating works as follows:

- If the call is not established, they rate the call 0 stars.
- If the call is established without delay, the initial rating is 5 stars (the maximum).
- For every 10 units of the Euclidean distance between the customer and the selected POP, a star is lost:

```
dist = sqrt(pow(x_c - x_pop, 2) + pow(y_c - y_pop, 2))
stars = 5 - int(dist / 10.0)
```

- Customers don't move during a call, so distance is constant during a call's lifetime.
- If the call establishment is delayed, one star is lost per ten units of time (any delay incurs a loss of at least one star):

```
stars = stars - int(ceil(delay / 10.0))
```

- The minimum rating is always 0 stars:

```
stars = max(stars, 0)
```

So, for each call you have to decide whether to route it to a POP or not and whether you want to enqueue it or not. You have to explicitly indicate when a call will be established, even when it is delayed. A call cannot be scheduled to be established in a POP that has reached the limit.

Input

The first row contains two positive integers, P and K , which indicate the number of POPs and Calls that will be made.

P rows follow, where each row has the following format:

$X \ Y \ C$

- X is the integer with the X coordinate of the POP.
- Y is the integer with the Y coordinate of the POP.
- C is the positive integer with capacity of the POP, that is, the number of concurrent calls it supports.

K rows follow, where each row has the following format:

$X \ Y \ T \ D$

- X is the integer with the X coordinate of the call.
- Y is the integer with the Y coordinate of the call.
- T is the non-negative integer with the time when the call is made.
- D is the positive integer with the duration of the call.

Output

The output consists of the information about the calls that can be established. There may be calls that are unable to be established, which is the default action if they do not appear in a row.

For every call that is established, a row should be written with the following format:

$C \ P \ T$

- C is the non-negative integer with the index of the call we are going to establish (0 indexed).

- P is the non-negative integer with the index of the POP that will handle the call (0 indexed).
- T is the non-negative integer with the time at which the call will be established (which may be greater or equal to the time when the call is made).

Limits

- $1 \leq P, C \leq 20$
- $1 \leq K, T, D \leq 30000$
- $-100 \leq X, Y \leq 100$

Sample input

```
1 3
0 0 2
0 5 0 2
5 0 1 2
0 -30 1 3
```

In this problem we have 1 POP and 3 calls.

The POP is located at position (0, 0) and has capacity for 2 calls.

Calls are made from positions (0, 5), (5, 0) and (0, -30), each of which are at a distance of 5 to the POP except the last one, which is at a distance of 30.

The first call starts at time 0 and has a duration of 2, the second starts at time 1 and has a duration of 2 and the third starts at time 1 and has a duration of 3.

Sample output

```
0 0 0
1 0 1
2 0 2
```

All three calls are established.

The first row indicates that call 0 is routed to POP 0 at time 0, which occupies 1 of the 2 slots in the POP until time 2.

The second row indicates that call 1 is routed to POP 0 at time 1, which occupies 1 of the 2 slots in the POP until time 3.

The third row indicates that call 2 is routed to POP 0 at time 2 (with a delay of 1,

since the POP is full until time 2), which occupies 1 of the 2 slots in the POP until time 5.

Sample Score

This solution gets a score of 11, which happens to be optimal for this problem:

- Call 0 gets 5 stars, since it's established at distance 5 with no delay.
- Call 1 gets 5 stars, since it's established at distance 5 with no delay.
- Call 2 gets 1 star, since it's established at distance 30 (losing 3 stars) with a delay of 1 (losing an additional star).

There may be different solutions that will achieve the same score.

Test your code

You can test your program against both the input provided in the test phase and the input provided in the submit phase. A nice output will tell you if your program got the right solution or not. You can try as many times as you want to. Be careful with extra whitespaces, the output should be exactly as described.

Test your program against the input provided in the test phase

[Download test input](#)

Program output:

Seleccionar archivo Ningún archivo seleccionado

Submit test output

Test your program against the input provided in the submit phase

[Download input](#)

Program output:

Seleccionar archivo Ningún archivo seleccionado

Submit output

During the submit phase, in some problems, we might give your program harder inputs. As with the test token, a nice output will tell you if your program got the right

solution or not. You can try as many times as you need.

In the actual contest you first need to solve the test phase before submitting the code, you must provide the source code used to solve the challenge and you can only submit once (once your solution is submitted you won't be able to amend it to fix issues or make it faster).

If you have any doubts, please check the [info section](#).

« Prev Next »

Tweet about this! [#TuentiChallenge6](#)

[Follow @TuentiEng](#)