



Level 5



Glados:

Well, that wasn't great.

Our initial algorithms did pretty badly in some cases - we need our tower placement to be perfect. You have to create more powerful algorithms to place these towers, otherwise we're doomed! Oh, and our budget is kinda slim... And I'm sorry for calling you a greeny!

Task for Level 5:

Given a predefined number of towers, find the best place to put them on the map



- › Towers cost gold according to their type. The amount will be given in the input
- › **Towers cannot be built on the path** and **one cell can contain MAX one tower**
- › When you choose to build a tower, the amount of gold you have decreases by its cost
- › After you choose where to build your towers, the simulation behaves exactly as in level 4
- › After you've done enough thinking and found the solution, just send us the list of positions for the towers
- › You cannot change your mind during the simulation, once built, the towers cannot be moved or destroyed
- › You can't use more gold than you have, but you can choose to use less
- › To keep the simulation from becoming overly complex, **you are not allowed to build more than 500 towers**

> Level 5

Input format:

<WX> <WY>

<X> <Y>

<Movement_Sequence>

<Health> <Speed>

<N>

<SpawnTime>

...

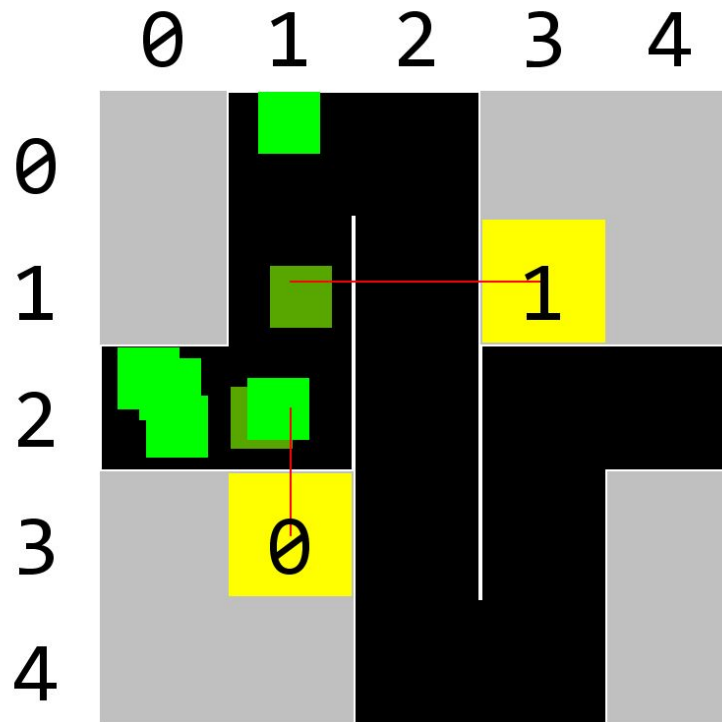
<SpawnTime>

<Damage> <Range> <Cost>

<Gold>

N times

-> the amount of gold you start with

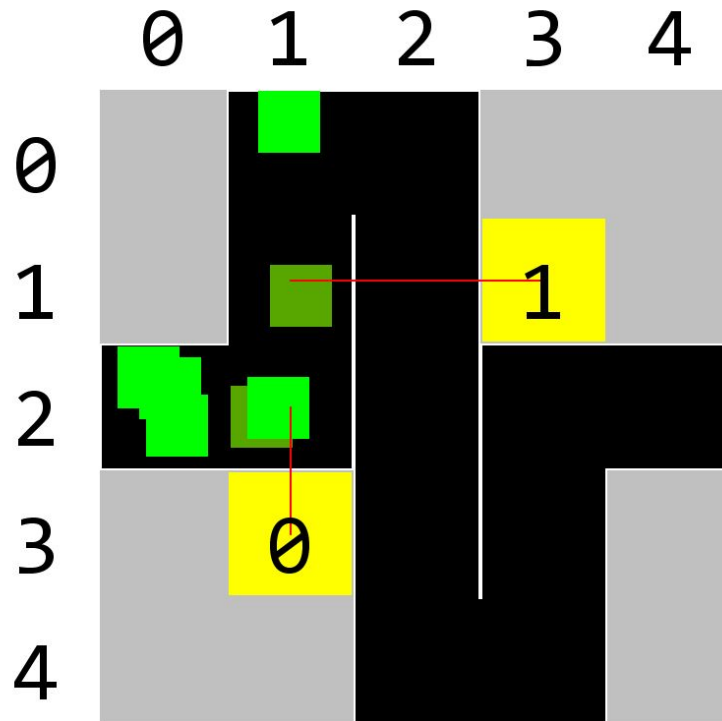


*Click to see animated version

Input example:

```
5 5
0 2
F 1 T 3 F 2 T 1 F 1 T 1 F 4 T 3 F 1 T 3 F 2 T 1 F 1
43.33 1.08
10
5
4
3
4
2
1
5
6
5
2
15.0 4 10
20
```

-> towers deal 15 damage, have a range of 4 and cost 10 gold
-> you have 20 gold to build towers



*Click to see animated version

> Level 5

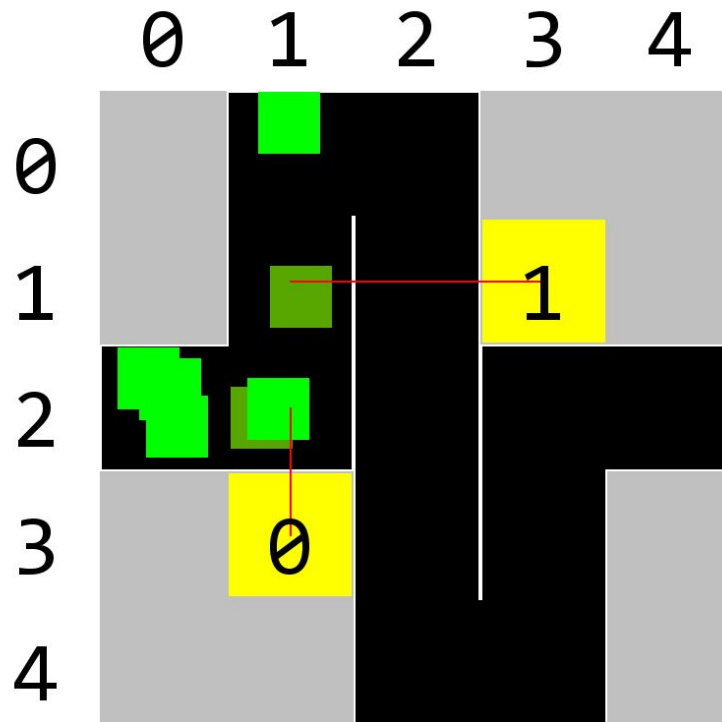
Output format:

<X> <Y>

<X> <Y>

...

Any configuration that manages to win is considered correct



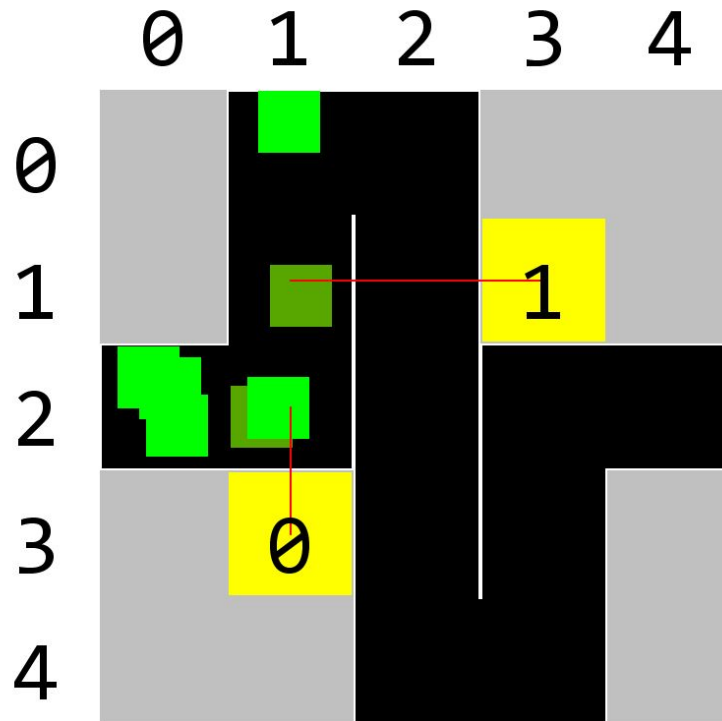
*Click to see animated version

> Level 5

Output example:

1 3

3 1



*Click to see animated version