Greedy Assumption 1: As we need to prevent the monsters from reaching our city by killing one monster every second, the greedy choice will be to kill the monster whose current position is closest to our city.

Proof of failure of above Assumption:

- We must realise that this greedy **assumption fails**, as, although a monster might be at a farther position than others at the current second, it might have such a **high speed** that at the very next second it reaches our city.
- Example: dist:[2,3], speed[1,5] --> If we choose to kill 0th(0-indexed) monster at time = 0 as it has a lower distance value, then at time = 1, 1st monster will already have reached our city so we'll only be able to kill a a total of one montser.
- Now instead, if we had killed the 1st monster first, then, at time = 1, 0th monster would still be at dist = 1 from the city, so we can kill it too and hence we'll be able to **kill both the monsters.**

Greedy Assumption 2:

- So, as our greedy position assumption 1 clearly fails, what shall we base our order of choices on? Something that will take care of both distance and speed? **TIME!**
- So, to bring some logical ordering in our approach, we use given initial dist[i] and speed[i] to calculate initial time[i].
- Logically, a monster with a lower time value would reach the city before a monster with a higher time value.

Time Calculation:

- As each monster much be killed at an inital second, so for example if dist = 1 , and speed = 2 , although time will be = 1/2, which in integer will give 0, we can still kill this monster, as at time =0, this monster will be 1/2 time units away from arrival.
- Conclusion --> Formula for time must be time[i] = ceil(dist[i]/speed[i]);

Algorithm:

- The last thing that's left now is pushing all these time values into a a min priority queue (min_heap) and greedily killing the monster with the lowest time of arrival value at each second and then popping it out of the heap.
- If at any instant, the element at the top of the heap, that is, the lowest time value is less than or equal to our curent_time instant value, this means that the monster has already reached our city and hence we can't kill anymore monsters.
  Time Complexity: O(NlogN + N) ~ O(NlogN): N for max number of seconds and NlogN for N heap operations with each operation taking logN time.