

## 789. Escape The Ghosts

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Solution](#)

---

You are playing a simplified Pacman game. You start at the point  $(0, 0)$ , and your destination is  $(\text{target}[0], \text{target}[1])$ . There are several ghosts on the map, the  $i$ -th ghost starts at  $(\text{ghosts}[i][0], \text{ghosts}[i][1])$ .

Each turn, you and all ghosts simultaneously *may* move in one of 4 cardinal directions: north, east, west, or south, going from the previous point to a new point 1 unit of distance away.

You escape if and only if you can reach the target before any ghost reaches you (for any given moves the ghosts may take.) If you reach any square (including the target) at the same time as a ghost, it doesn't count as an escape.

Return True if and only if it is possible to escape.

**Example 1:**

**Input:**

```
ghosts = [[1, 0], [0, 3]]
```

```
target = [0, 1]
```

**Output:** true

**Explanation:**

You can directly reach the destination  $(0, 1)$  at time 1, while the ghosts located at  $(1, 0)$  or  $(0, 3)$  have no way to catch up with you.

**Example 2:**

**Input:**

```
ghosts = [[1, 0]]
```

```
target = [2, 0]
```

**Output:** false

**Explanation:**

You need to reach the destination  $(2, 0)$ , but the ghost at  $(1, 0)$  lies between you and the destination.

**Example 3:**

**Input:**

```
ghosts = [[2, 0]]
```

```
target = [1, 0]
```

**Output:** false

**Explanation:**

The ghost can reach the target at the same time as you.

**Note:**

- All points have coordinates with absolute value  $\leq 10000$ .
- The number of ghosts will not exceed 100.

---

Seen this question in a real interview before?

Firstly, you calculate `max`, which is the Manhattan distance of the starting point to the target;  
Next, for each of the `ghosts`, you calculate the Manhattan distance between that ghost and the target;  
If this distance is smaller, then it means that this ghost would eat the pacman (because he would reach the target *before* the pacman) and hence we return `false`; else we return `true`.

Good work!

```
class Solution {
public:
    bool escapeGhosts(vector<vector<int>>& ghosts, vector<int>&
target) {
        int maxdist = abs(target[0]) + abs(target[1]);
        for(auto elem:ghosts)
        {
            int dist = abs(elem[0]-target[0]) + abs(elem[1]-
target[1]);
            if(dist<=maxdist)
                return false;
        }
        return true;
    }
};
```