Moon

Problem D

Chin Ka Wang {rina__owo}

The 1st Buddhist Sin Tak College Computer Club Programming Contest

October 18, 2024

stpc();

Background

Problem Idea by `rina__owo`
Preparation by `rina__owo`, `pepper1208`

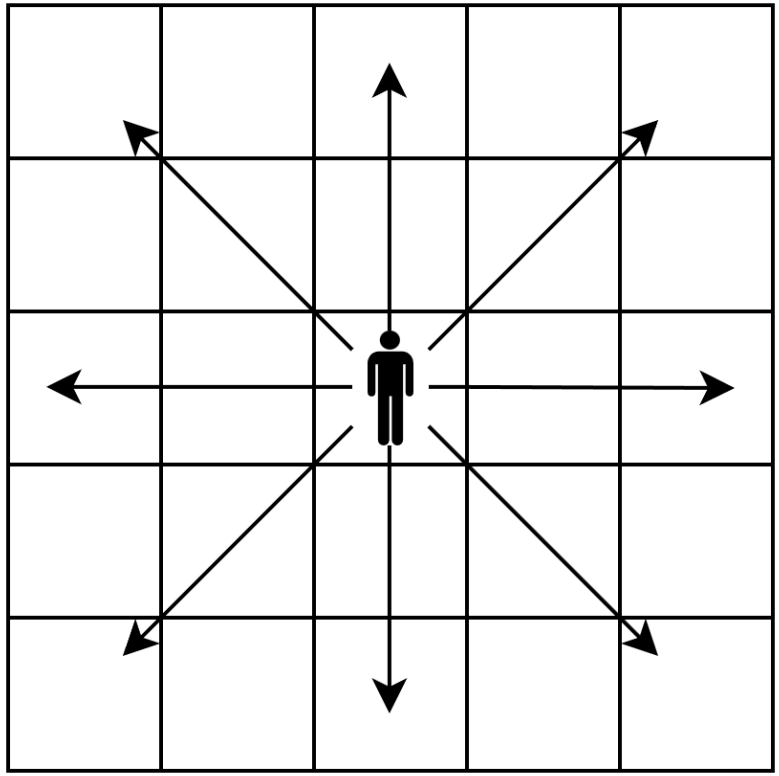
Problem Restatement

Given an $N \times M$ grid. Each cell may contain a building, a person, or nothing. A moon can be placed in any cell and replaces the content of that cell.

Each person can only view the moon in eight straight-line directions: East, South, West, North, Northeast, Northwest, Southeast, and Southwest. If a person cannot see the moon from any direction, or their line of sight is blocked by a building, they are considered unable to see the moon. People do not block the view of other people.

Find the coordinates of the cell where the moon should be placed so that it can be seen by the most people. If there are multiple possible coordinates, output the lexicographically smallest coordinates.

Problem Restatement



Statistics

Points are given per subtask in this problem. There are 7 subtasks in this question.

Attempts: 9

0 points	5	+	1	=	6
Subtask 1 (5 points)	0	+	0	=	0
Subtask 2 (6 points)	0	+	0	=	0
Subtask 3 (13 points)	0	+	0	=	0
Subtask 4 (8 points)	0	+	0	=	0
Subtask 5 (14 points)	0	+	0	=	0
Subtask 6 (23 points)	0	+	0	=	0
Subtask 7 (31 points)	0	+	0	=	0

First solved by **No one**!

Subtasks

Let H be the number of Dr. Jones's friends, and B be the number of buildings.

Subtask	Score	$N \times M \leq$	$H \leq$	$B \leq$
1	5	10	1	0
2	6	100	1	0
3	13	100	1	1
4	8	100	2	0
5	14	100	2	2
6	23	100	$N \times M$	2
7	31	10^4	$N \times M$	$N \times M$

Solution

This question is a large-scale **simulation** question.

Note that each person can only view the moon in eight straight-line directions. We can implies that the moon can receive sight from that eight directions as well.

The question then can transformed into, find the position where the moon can receive sights from the most people.

We can simply achieve this through enumerating each position of the moon and store the largest number of sights the moon can receive among all situations.

Subtask 1 (5 pts)

Subtask 1 ($N \times M \leq 10$, $H = 1$, $B = 0$)

Set up an array `arr` to store the input grid and set up a two-layer `for loop` to iterate every cell of `arr` for the moon to put onto.

In every cell `arr[i][j]`, set up 8 `for loop` or `while loop` (depends on which direction) to iterate all cells in 8 directions with `arr[i][j]` as the centre in order to count the number of people lie on the 8 directions. Sum of the number in all directions and store the result of all position of moon in the corresponding cell in another array `res`.

The iteration on each direction should end when the cell in the next iteration is out of the grid.

Subtask 1 (5 pts)

Set up a `for loop` to iterate all the cells in `res` first by *i* then by *j* and store the largest number in a variable `largest`, and the corresponding coordinates in an array `ans` with two cells. (`ans[0]` stores *i* and `ans[1]` stores *j*)

Whenever a larger number is detected, update the value of `largest` and `ans`. If a number with the same value as `largest` is detected, do not update the value of `ans` as its lexicographically coordinates must be larger than the one in `ans`, due to our iteration order.

Score: 5

Subtask 2 (6 pts)

As $B = 1$ in task 2, we have to now consider the "blocking" of buildings.

The solution of task 2 is basically same as task 1, besides the iteration on each direction should end when a building is detected.

Score: 11 (Task 1 + Task 2)

Full Solution

By implementing the algorithm in subtask 2, you can AC the problem easily.

Score: 100

Takeaways

1. A solid coding skills must be required for large-scale simulation problem.
2. Be familiar with decomposing the problem into several section that helps you to get in touch with the macroscopic view of the problem.