香港電腦奧林匹克競賽
Hong Kong Olympiad in Informatics

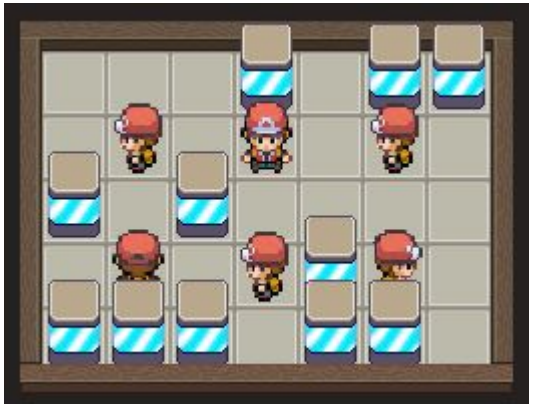
S241 - Mirror Horror

Ethen Yuen {ethening}
2024-02-17

Background

Problem Idea by kctung

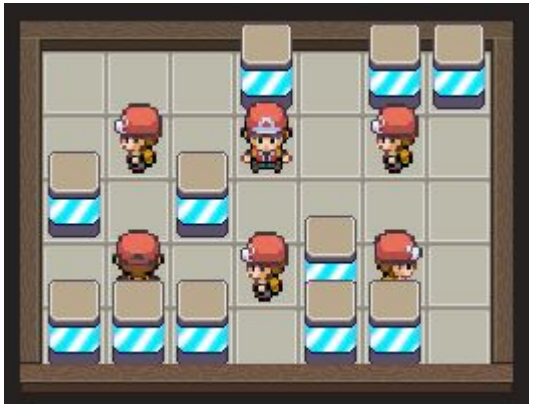
Preparation by mlwong, ethening, kobebryan9



Problem Restatement

Given a (2N+1) x (2M+1) grid with people in some cells (even row & col) and mirrors in some cells.

Task: Choose a direction for all people to face, such that they don't look into each other and mirror. *Some directions are predetermined



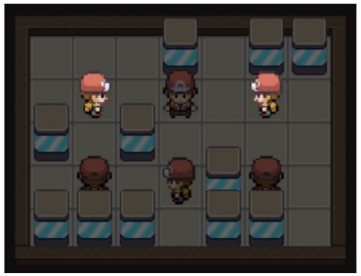


Figure 2: People making eye contact. Note that the person between them will not obstruct their view.

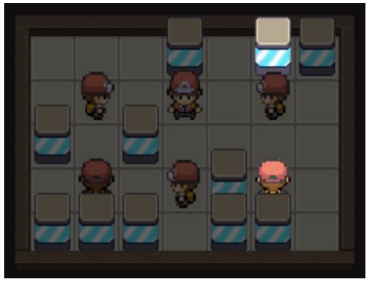


Figure 3: A person facing a mirror. Again, the person between him and the mirror will not obstruct his view.

Figure 4: People feeling comfortable.

2 3 ...M.MM .?.v.<. M.M.... .^.<M?. MMM.MM.	Possible ...M.MM .<.v.<. M.M.... .^.<M>. MMM.MM.

Statistics

0 points	6 + 0 + 0 + 0 =	6
11 points	3 + 0 + 0 + 0 =	3
12 points	5 + 0 + 0 + 0 =	5
21 points	3 + 0 + 0 + 0 =	3
23 points	0 + 1 + 0 + 0 =	1
32 points	2 + 0 + 0 + 0 =	2
38 points	7 + 1 + 0 + 0 =	3
49 points	2 + 6 + 1 + 0 =	9
60 points	1 + 0 + 0 + 0 =	1
71 points	5 + 4 + 0 + 0 =	9
75 points	0 + 1 + 0 + 0 =	1
86 points	1 + 0 + 0 + 0 =	1
100 points	0 + 8 + 13 + 8 =	29

First solved by **yellowtoad** at **15m 1s**

Attempts	Max	Mean	Std Dev
78	100	60.871	36.237

Subtasks						
12: 69	9: 63	11: 54	17: 58	22: 41	15: 31	14: 29

Subtasks

If i and j are both even, then $C_{i,j}$ must be one of `^`, `v`, `<`, `>` or `?`. `^` means the person in the cell must face up, `v` means the person in the cell must face down, `<` means the person in the cell must face left, `>` means the person in the cell must face right, and `?` means that the direction has not been decided for the person in the cell.
Otherwise, $C_{i,j}$ must be `M` or `.`, where `M` means the cell contains a mirror block, and `.` means the cell is empty.

SUBTASKS

	Points	Constraints
1	12	$1 \leq N, M \leq 250$ Everyone's direction has not been decided. There are no mirror blocks.
2	9	$1 \leq N, M \leq 250$ Everyone's direction has not been decided. There is exactly one mirror block.
3	11	$1 \leq N, M \leq 250$ Everyone's direction has been decided by Bob.
4	17	$1 \leq N, M \leq 250$ Everyone's direction has not been decided. There is at most one mirror block in each row. There is at most one mirror block in each column.
5	22	$1 \leq N, M \leq 250$ Everyone's direction has not been decided.
6	15	$1 \leq N, M \leq 1000$ Everyone's direction has not been decided.
7	14	$1 \leq N, M \leq 1000$

Subtask 1

Subtask 1 (12%): All directions not decided, No mirror blocks

- Sanity check. Simply output a config where no one face each other.
- A easy answer is to make everyone face the same direction.

Score: 12

Time Complexity: O(NM)

..... .?.?.?. ..... .?.?.?.	..... .<.<.<. ..... .<.<.<. .....

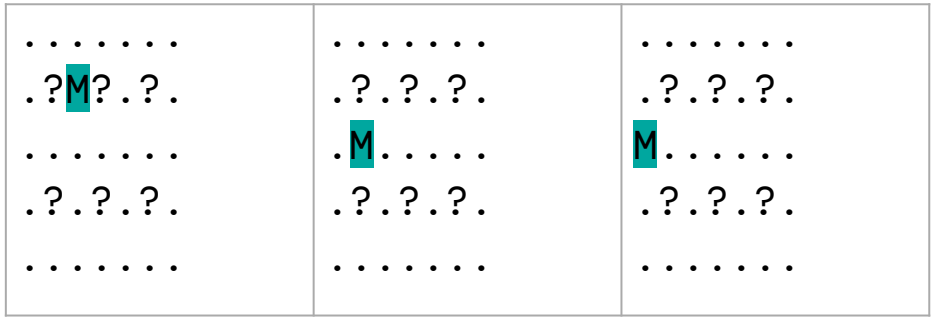
Subtask 2

Subtask 2 (9%): All directions not decided, Exactly one mirror block

- The mirror block only affect the directions when it is on the same row of some people (config 1) / on the same column of some people (config 2).
- Either make everyone face up, or make everyone face left. One of these always works.

Score: 9 (Cumulative: 21)

Time Complexity: O(NM)



Subtask 3

Subtask 3 (11%): All directions are decided, N, M <= 250

- This subtask requires you to write a simple checker.
- The most naive way would work: start from each person and loop through cells in the direction it is facing.
  - Invalid when you meet cells facing the opposite direction or mirror.

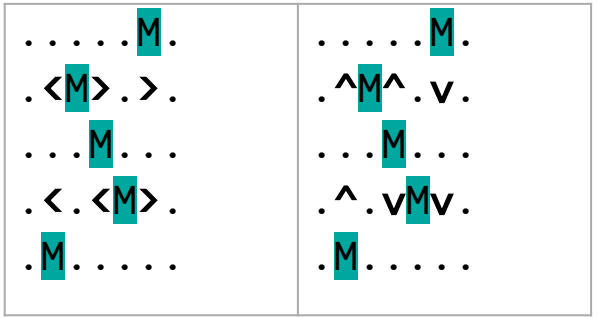
Score: 11 (Cumulative: 32)

Time Complexity: O(N^2 M^2)

Subtask 4

Subtask 4 (17%): All directions not decided, At most one mirror block per row/col

- Most important subtask for observation.
- Consider how to solve a single row: Each row has at most one mirror, let's make every person on left face left, and those on right face right.
- Solve each row one by one with this algorithm!
- (This also works similarly with column)



Score: 40 (Cumulative: 49)

Time Complexity: O(NM)

Subtask 5 & 6

Subtask 5 & 6 (22% & 15%): All directions not decided, (N, M <= 250 / N, M <= 1000)

- Think about what cannot be handled by subtask 4's solution: What will happen when a row has >= two mirror block?

```
.?M?.?.?M?.?M?.?.
```

Subtask 5 & 6

Subtask 5 & 6 (22% & 15%): All directions not decided, (N, M <= 250 / N, M <= 1000)

- Think about what cannot be handled by subtask 4's solution: What will happen when a row has >= two mirror block?

```
.<M?.?.?M?.?M>.>.
```

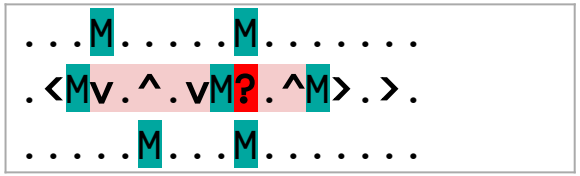
- While the people on two sides can be solved easily, the people in the middle can neither face left nor right no matter what the configuration of other cells are.
- The same applies to column, the middle part can neither face up or down.

```
.
^
M
?
M
V
.
```

Subtask 5 & 6

Subtask 5 & 6 (22% & 15%): All directions not decided, (N, M <= 250 / N, M <= 1000)

- If any person is surrounded by mirror on all sides



  It can face no direction at all -> impossible
- Consider the possible case: process the rows and columns separately
  - For each row, make every person on left of first mirror face left, and those on right of the last mirror face right.

Subtask 5 & 6

Subtask 5 & 6 (22% & 15%): All directions not decided, (N, M <= 250 / N, M <= 1000)

- Consider the possible case: process the rows and columns separately
  - For each row, make every person on left of first mirror face left, and those on right of the last mirror face right.
  - For each column, do the same.
- Why can we handle both directions separately?

Subtask 5 & 6

Subtask 5 & 6 (22% & 15%): All directions not decided, (N, M <= 250 / N, M <= 1000)

- Consider the possible case: process the rows and columns separately
  - For each row, make every person on left of first mirror face left, and those on right of the last mirror face right.
  - For each column, do the same.
- Why can we handle both directions separately? People facing left / right will never be affect by people facing up / down. **They are independent**.
- The impossible case can be distinguished by checking if there is still a “?” after the process.

Score: 60 / 75 (Cumulative: 71 / 86)

Time Complexity: O(N^2 M^2 / NM) < Depends on how you implement this

Full Solution

Subtask 7 (14%): No additional constraints

- What cases become unable to handle with subtask 6's solution?

```
.?M?.?.?M?.?.>.?.
```

- This is ok

```
.?M?.?.?M?.?.v.?.
```

- This is ok too.

Full Solution

Subtask 7 (14%): No additional constraints

- What cases become unable to handle with subtask 6's solution?

```
.?M?.?.?M?.?.<.?.
```

- This is not ok (The < breaks the whole config no matter what)

```
.?.?.?.?.<.?.>.?.
```

- This is not ok. (No mirror but you cannot place > before the < in the input)

Full Solution

Subtask 7 (14%): No additional constraints

- The key part is: On each row, preplaced < and > also constrained your placement of < and >. (Like how a mirror constrained the placement)
- Revise the strategy for each row:
  - From left to right, place < at ? positions until you meet > or M.
  - From right to left, place > at ? positions until you meet < or M.
- Do the same for each column.
- Then done!

Full Solution

Subtask 7 (14%): No additional constraints

- Then done?
- You still need to check if the whole configuration is fully placed (no more ?) and if it is valid as a whole.
  - Because preplaced direction may already ruined the config.
  - You can improve subtask 3 checker by checking each row/col in a batch.

Score: 100

Time Complexity: O(NM)

Conclusion

- Some takeaways
  - Don’t be scare away simply because the statement is long or because it “looks complicated”
  - Observe what elements are independent / dependent on other elements, try to simplify the problem with it
    - Turning 2D problems into 1D problems is a common way
  - Subtask is used to guide your thinking
    - Think about what cases you cannot handle with previous subtasks’ solution is a good way