Find the Box

EGOI 2023

Problem author: Nils Gustafsson.

The Problem

There is an $H \times W$ grid. In some unknown cell there is a box. Your goal is to find the box.

Every night, a robot starts in the top left corner, and moves around the grid. You can decide how the robot should move by giving it instructions in the form of a string consisting of characters `<`, `>`, `^`, `v`.

The walls are solid, so if the robot attempts to move outside of the grid, nothing will happen. The box is also solid, so you cannot move into the cell containing the box.

At the end of each night, the robot will report to you its location, and go back to the top left corner. Your task is to find the location of the box in at most Q nights. In particular, you get full score if $Q \leq 2$.

Solution 1: $Q = 50$ (20 points)

Notice that if the robot is on $(r, 0)$, and the box is on (r, c), where $c > 0$, you may instruct the robot to keep going right and it will eventually get stuck on $(r, c - 1)$.

On the other hand, if the box is not on row r, the robot reaches $(r, W - 1)$.

Therefore, you can brute force each row. For row r, move r times downwards, then move $W - 1$ times rightwards and see if the box is stuck.

But be careful of the special case when the box is in column 0.

When $H = 50$, this solution uses $H = 50$ queries, which scores 20 points.

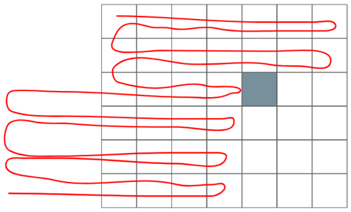
Solution 2: $Q = 3$ (82 points)

Using a similar idea to the previous solution, if we know which column the box is in, we can easily find its exact position by instructing the robot to go to the correct column, then moving down many times such that it gets stuck eventually.

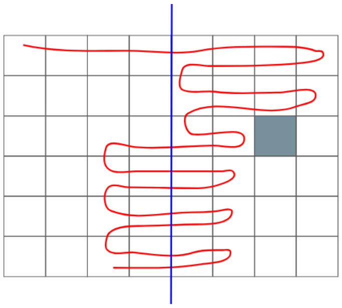
Therefore, we may consider focusing on finding the correct column of the box using 2 queries, then using 1 final query to find the position of the box. That adds up to 3 queries. Now, the question is how we can do it.

When encountering similar types of grid interactive problems, some common strategies contestants may consider include constructing some well-defined patterns (like spirals), or dividing the grid into parts, considering parities, randomized methods, etc. In fact, the first two methods are helpful with solving this task using $Q = 3$ queries!

Let's consider the following pattern: moving back and forth along each row (like in the figure below). When the robot encounters the box, we know that it gets stuck. As a result, the robot will be displaced to the left, revealing which column the box is in.

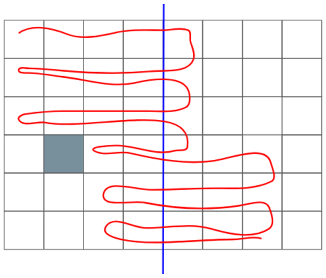


While this method seems promising, there is a small issue: the robot will run into the left wall and the information about the column will be lost. So instead, let's divide the grid into two halves and run this solution on both halves! The first query will look like this:



When the robot enters the second half of the grid, let's say it moves x units back and forth along each row. If the robot finally lands on $(H-1, c)$ then it must have got stuck somewhere at column $c+x$, so the box would be at column $c+x+1$.

Of course, the above solution assumes that the box is in the right half of the grid. If the box had been in the left half, we would need one more similar query to find the correct column:



There are also some special cases, like if the box is in the first row. But if you are careful enough, this will let you solve the problem in only three queries.

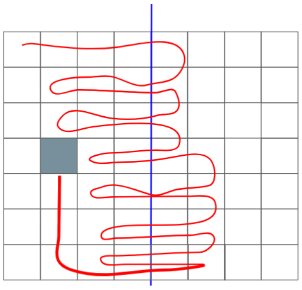
Solution 3: $Q = 2$ (100 points)

To get the number of queries down to 2, we will modify the $Q = 3$ solution.

The main idea is that we do not always need the final query to get the position given the correct column. Sometimes we can incorporate that into the query that finds the column.

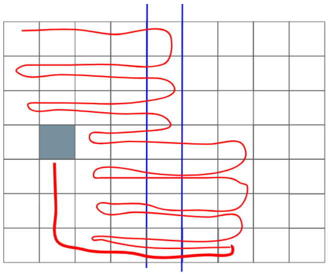
Let's start by making the same first query as in the $Q = 3$ solution. If the box was in the right half of the grid, we would find the column of the box using one query and find the row as well using another query. This takes 2 queries, so there is no issue.

The issue arises when the box was in the left half, we need to find it in only one more query. This is a bit tricky to do, and it is convenient to divide it up in two cases depending on whether W is even or odd. For even W, we can do this:



In other words, we find the column similar to how it was done in the previous solution, and then we also move back to the column and get the position of the box. This is possible thanks to the fact that we know that the box is not in the right half, so we can move freely there.

When W is odd, then this second query can look like this instead. Note that there are some special cases to be handled, such as when the box is in the middle column.



With careful handling, one may solve the problem using $Q = 2$ queries, which scores 100 points.

Aftermath

There are also many other ways to get full score on this problem. If you ask someone who solved it, then you will quite likely hear something different.

Can you prove that it is not possible to solve the problem with $Q = 1$?