Programming challenges in Java

5. Exploring algorithms (I). Brute force

Nacho Iborra

IES San Vicente

Table of Contents

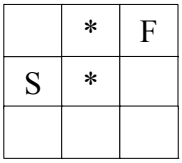
Programming challenges in Java

1. Brute force algorithms

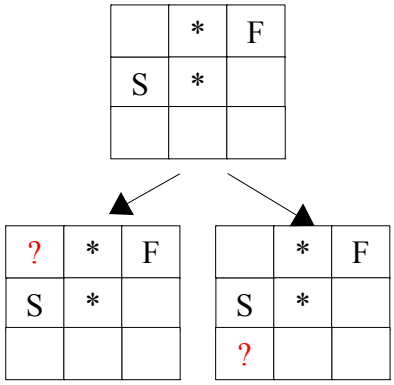
Brute force algorithms are algorithms that explore all the possible paths of a given problem in order to find a solution. Depending on the problem itself, we may need an array, list, multi-dimensional array... and often a recursive function to solve the problem.

Let's have a look at this challenge from Acepta el Reto.

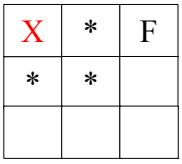
We need to explore a bidimensional map in order to find a path from the starting point (S) to the finish point (F). Let's suppose that we have a 3 x 3 map with this structure:



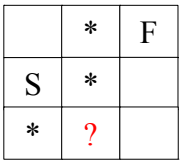
How could we check if there is a solution (a path from S to F)? We need to apply a brute force algorithm: starting from S, we need to explore all the contiguous cells that are empty (this is, the don't have a '*'). In this case, we have two possible options:



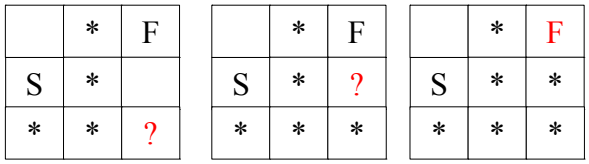
So we mark our cell as "visited" and check these two options: From the first one, there is no possible "next cell", so we discard this option:



Regarding the second path, we can go right:



And we can apply the same procedure until we find the way to the finish point:



We need to define a recursive function that takes the map as a parameter, and the current position of the explorer (row and column), then, it must explore all the possible next steps (up, down, left and right) and return *true* whenever the "F" point is reached.

```
public static boolean findExit(char[][] map, int row, int col)
{
    // If we found the "F", return true
    // Else, return if we find it going up, down, left or right
}
```

1.1. Try yourself: Escapando de las fuerzas imperiales

Try to apply this algorithm to this challenge and solve it.