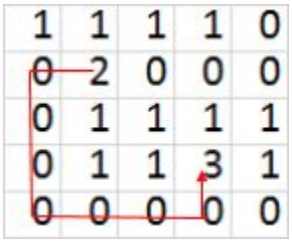
Lab 1: Search

The task in this lab is to create a search algorithm to help a robot find its way to a destination. The robot exists in a grid world named “map” of size MAP_WIDTH by MAP_HEIGHT, and it can move in all 4 directions (diagonals not allowed) through empty space cells only by steps of exactly 1 cell distance.
The starting location of the robot is marked in map with a number “2” and the goal with a number “3”. The other two values you can find in the map are “1” for walls and “0” for empty space.

Sample map of size MAP_WIDTH 5 and MAP_HEIGHT 5



Robot starting location in map[1][1]
Goal in map[3][3]

You are to complete the code for 2 functions: **df_search**(int **map) and **bf_search**(int **map). The first one uses depth first search and the second one uses breadth first search to find a path from the robot starting location to the goal. The functions should return a boolean value “true” if the destination was reached and “false” otherwise. An additional condition is to mark the map with a number “4” in all explored cells and with a number “5” in the cells that are part of path found.

To make sure everybody arrives to the same results (very important for the automated grader) you **must** use the following search order for map[y][x]:
First [y][x+1], then [y+1][x], then [y][x-1], and finally [y-1][x]

Considerations:

- We provided several maps to let you test your solution, but the grading will use a different set.
- Remember to use the provided constants for the map boundaries and do not hardcode any values because during grading the dimensions of the map can be different.
- The starting location of the robot and the destination are part of the path and should be marked with a “5” in the map.
- The running time of your algorithm cannot be longer than 5 seconds for a 15x15 maps or smaller, otherwise it will fail the grading tests.
- All maps will have a maximum of 1 possible path between the starting location and the destination (to make it easier).
- There will be no loops in the maps (to make it easier).