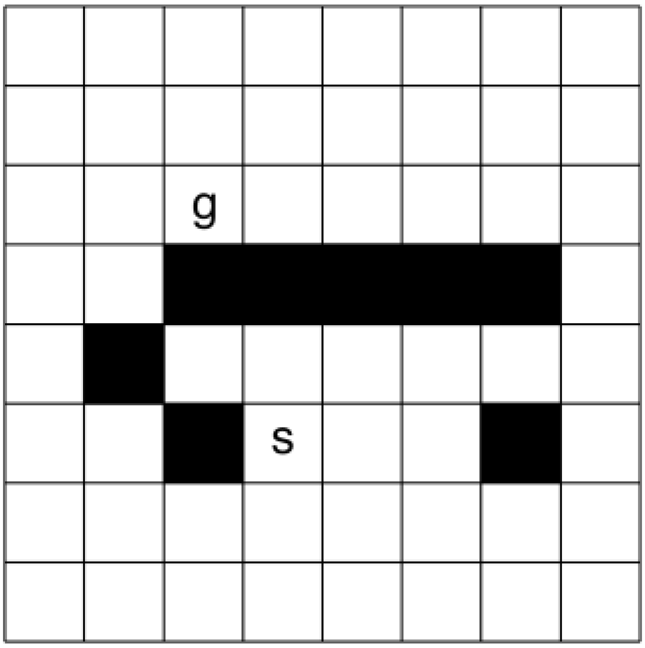
Assignment 2: Problem solving as search

Consider the problem of finding a path in the grid shown in the figure below from the position s to the position g. A piece can move on the grid horizontally or vertically, one square at a time. No step may be made into a forbidden black areas.



Choose one of the following cases to solve and write a python program for implementing it, showing how nodes are explored and how paths are expanded:

1. On the grid shown above, number the nodes expanded (in order) for a depth-first search from s to g, given that the order of the operators is up, left, right, and down. Assume there is cycle pruning. What is the first path found?
2. On a copy of the same grid, number the nodes expanded, in order, for a greedy best-first search from s to g. Manhattan distance should be used as the evaluation function. The Manhattan distance between two points is the distance in the x-direction plus the distance in the y-direction. It corresponds to the distance traveled along city streets arranged in a grid. Assume multiple-path pruning. What is the first path found?
3. On a copy of the same grid, number the nodes expanded, in order, for a heuristic depth-first search from s to g, given Manhattan distance as the evaluation function. Assume cycle pruning. What is the path found?
4. Number the nodes in order for an $A*$ search, with multiple-path pruning, for the same grid. What is the path found?

Note: 1- You need to plot the solution animated,
i.e. showing the movement of the agent passing from one cell to the other until it arrives to the goal cell.
2- You need to plot the graph that represent the problem space,
where nodes are the cells and the arcs are the actions (move_up, move_down, move_left,move_right),
which they might be associated with costs.