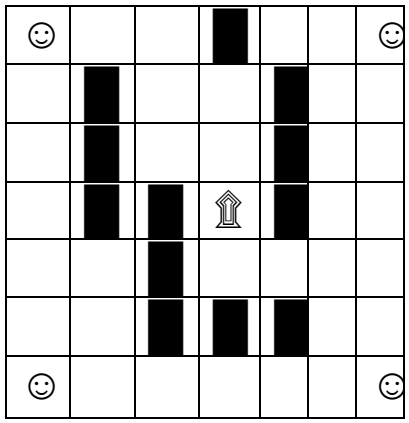
CSI4016 – Assignment 2
Winter 2015

Purpose:
The purpose of this assignment is to allow you to experiment with an informed search algorithm, which will include a representation of the world, an algorithm for search, and the observation of some metrics associated with the search algorithms you are using.

Description:
Consider the following maze in which each smiley face indicates a possible starting position, and the middle square containing a house represents the goal position. The squares containing black blocks represent obstacles that cannot be crossed by a smiley face. Smiley faces cannot go outside of the bounds of the maze either.

The smiley faces are allowed to move in 4 directions: up, down, left and right, as long as no obstacle blocks their way. They can only move by one square at a time.

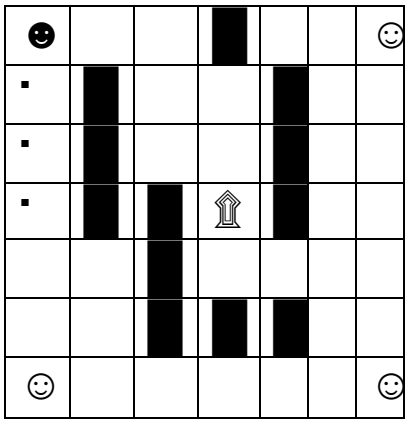


Write a Java program that simulates a race of the four smiley faces, assuming that they each use the A* algorithm along with the following heuristic function, h'(x):

(1) Number of vertical squares necessary to reach home from the starting position assuming that no obstacles were present
+
(2) Number of horizontal squares necessary to reach home once the vertical distance to home has been travelled, assuming that no obstacles were present
+
(3) Number obstacles encountered in the two previous calculations.

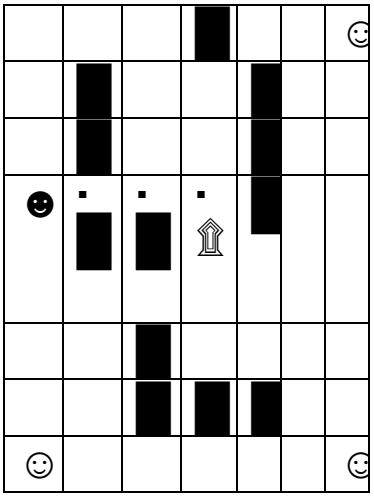
In other words, A* will use the usual g(x) function, describing the actual cost of reaching node x and h'(x), given above, describing the optimistic estimate of h(x), where h(x) is the actual *unknown* remaining cost.

Estimated total cost from node x to goal f(x) = g(x) + h'(x).



An illustration of what h'(x) computes:

Step (1): 3 vertical squares were travelled



Step (2): 3 horizontal squares were travelled

Step (3): 0 obstacle was encountered in Step (1) and 2 obstacles were encountered in Step (2)

h'(x) = 3 + 3 +2 = 8 and g(x) = 0, since we considered the smiley's starting position (distance from starting node to the current node.

- Note 1: The race does not need to be executed in parallel. Complete the race with smiley 1, first, record the results, and repeat with smileys 2, 3 and 4.
- Note 2: You will need to represent the maze, including the location of the smiley face, home, and the obstacles.
- Note 3: Your moves will need to check whether the position you are considering is empty or has an obstacle in it prior to execute. A smiley is not allowed to land on a square containing an obstacle.
- Note 4: Please, note that the number of nodes visited is different from the number of positions visited on the path from the starting position to home.

Your program should take as input the starting positions of each smiley face and should output the list of positions traveled from that starting position to the goal, for each smiley face, along with the length of that path. It should state the winner of the race by

comparing the length of the path. At the end you should output the winner of the competition.

Repeat the same exercise with the following maze in place (note the extra obstacle that has been added).