

# Algorithms for Game Design

## Exercise #4

To be completed after Session 4, before Session 5

### 1) Textbook

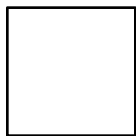
If you haven't already, read Chapters 0 - 3 and 5.

Then, read Chapter 4: Game AI: Collisions.

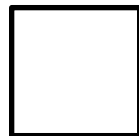
### 2) Navigation Algorithms

Let's check that you understand each of the navigational algorithms we talked about in Session 4. Here is a map, with some terrain costs. Use it for the following questions.

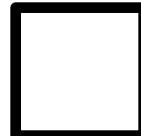
In any case where you need to break ties, pick the northmost tile first. If both tiles are in the same row, pick the westernmost tile first.



= 1



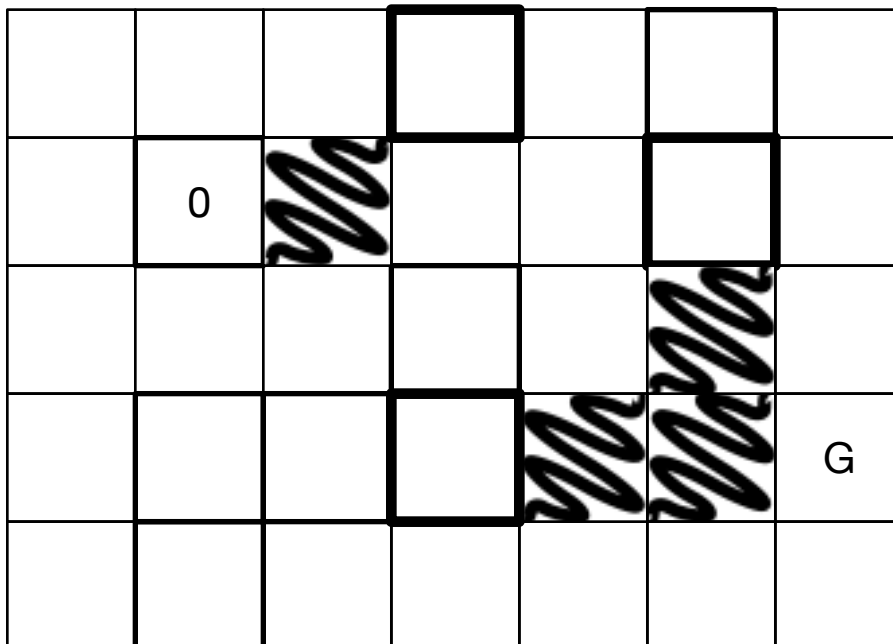
= 2



= 3



Obstacle



Part A) Use the BFS algorithm, as specified in the code in class, to search from the '0' tile to the 'G' goal. On the grid, write a number in each tile to show the order in which it is drawn from the frontier.

Part B) Use Dijkstra's algorithm to do the same search. How is this different from a BFS?

Part C) Use GBFS to search from 0 to G. Again, you should know how it is different from BFS or Dijkstra's. On the grid, write an arrow to show the "came\_from" arrow when the cell is added to the frontier. If the cell ends up in the frontier multiple times, show all the arrows.

Part D) Use A\* to repeat the search. Draw the "came\_from" arrows. Again, you should be able to describe the difference between A\*, GBFS, Dijkstra's and BFS.