**Project 1: The Searching Pac-Man**

Rishi Josan – 108996773, Abhiroop Dabral – 108200755, Ankit Dave – 109364245

### Depth First Search

**TinyMaze**

Total Cost: 10

Number of Nodes Expanded: 15

Score: 500

Running Time: 0

**Medium Maze**

Total Cost: 170

Number of Nodes Expanded: 273

Score: 380

Running Time: 0

**Big Maze**

Total Cost: 210

Number of Nodes Expanded: 646

Score: 300

Running Time: 0

Depth first search does seem to be a reasonable solution, though not the best. Not all the nodes that are visited are actually traversed. The solution is not the least cost solution though. There are two issues. The first one being, DFS by nature goes deep and consequently may end up going deep on the wrong path. The second is, we have no notion of how to rank successive nodes.

### Breadth First Search

**Medium Maze**

Total Cost: 68

Number of Nodes Expanded: 273

Score: 442

Running Time: 0

**Big Maze**

Total Cost: 210

Number of Nodes Expanded: 646

Score: 300

Running Time: 0

BFS provides the least cost solution to the goal

### Uniform Cost Search

**Medium Maze**

Total Cost: 68

Number of Nodes Expanded: 271

Score: 442

Running Time: 0

**StayEast Search Agent**

**Medium Dotted Maze**

Total Cost: 1

Number of Nodes Expanded: 187

Score: 646

Running Time: 0

**Medium Scary Maze and Stay West Search Agent**

Total Cost: 68719479864

Number of Nodes Expanded: 109

Score: 418

Running Time: 0

All the cases run successfully and we do get very low(1) and very high path(68719479864) costs for the StayEastSearchAgent and StayWestSearchAgent respectively

### A\* Search

**Big Maze and Manhattan Heuristic**

Total Cost: 210

Number of Nodes Expanded: 550

Score: 300

Running Time: 0

**Big Maze and Null Heuristic**

Total Cost: 210

Number of Nodes Expanded: 623

Score: 300

Running Time: 0

**Open Maze and Manhattan Heuristic**

Total Cost: 54

Number of Nodes Expanded: 91

Score: 456

Running Time: 0

**Open Maze and Null Heuristic (Same as BFS)**

Total Cost: 54

Number of Nodes Expanded: 683

Score: 456

Running Time: 0

**Open Maze and DFS**

Total Cost: 158

Number of Nodes Expanded: 683

Score: 352

Running Time: 0.2

We see that both in open and big maze for the Null heuristic, a large number of nodes are visited as compared to when we use Manhattan distance. A\* works well here.

Taking the Null Heuristic is the same as performing BFS. What is noteworthy is that DFS performs poorly here. A zig zag path is traversed which is far from optimal

### Finding Corners

**Tiny Corners**

Total Cost: 40

Number of Nodes Expanded: 390

Score: 501

Running Time: 0

**Medium Corners**

Total Cost: 135

Number of Nodes Expanded: 2870

Score: 406

Running Time: 0

For state in the corners problem, we append the list of corners in the start state. Everytime getSuccessors is called, we verify if the node just visited was a corner or not. If it is a corner, we update the current state by removing said corner from the list. The goal is reached when the corner list is empty.

### Corner Heuristic

Total Cost: 108

Number of Nodes Expanded: 788

Score: 433

Running Time: 0

### Food Heuristic

Total Cost: 60

Number of Nodes Expanded: 5417

Score: 570

Running Time: 4.3

We employ a very simple food heuristic that works very well. The heuristic is simply the sum of Manhattan distances to all the food nodes. This heuristic is admissible. It also runs quickly and very few nodes expanded. We had thought of using the Manhattan distance to the farthest node as the heuristic, but that performed poorly with about 10,000 nodes expanded.