Project 1 Search

10.18.22 Honghong Fang

1: Finding a Fixed Food Dot using Depth First Search

Pacman actually does not go to all the explored squares on his way to the goal. In DFS, we try to go as deep as possible first. To implement this, I use a stack to represent the frontier. We can see that our "explored set" needs only to be bd rather than b^d. However, DFS may not be the best one.

2: Breadth First Search

BFS does not find a least cost solution. The idea of BFS is to search the closest nodes first. To implement it, I use a queue to represent the frontier. However, the problem with BFS is that it is exponential in time and space. Exponential time is not as large of a problem as exponential space, as we quickly run out of memory. We can see that in order to expand all the nodes down to our solution depth, d, we would have to expand b^d nodes, where b is the branching factor.

3: Varying the Cost Function

How does the cost function influence BFS?

Best First Search is an informed search algorithm. It uses a heuristic function for deciding the expanding node. The function used in BFS is of the type f(n)=g(n)+h(n), where h(n) is the heuristic function and g(n) is the actual cost from starting node to node n. We can improve upon best first search by taking into account the cost used to get to a node as well as the predicted cost.

4: A* search

What happens on openMaze for the various search strategies?

	COST	TIME	NODES	SCORE
DFS	298	0.1	576	212
BFS	54	0.1	682	456
UCS	54	0.2	682	456
ASTAR	54	0.1	535	456

Compared with other strategies, DFS has the highest cost and the lowest score, but the number of expanded nodes is not the largest. Although BFS and UCS have the same cost as ASTAR, they have the largest number of expanded nodes. The optimal case is ASTAR with heuristic function added, which has the lowest cost. In the case of the minimum number of nodes.

5: Finding All the Corners

What is the advantage of using A* **over BFS or DFS?**

DFS is good because it allows a solution to be found without expanding all competing

branches. BFS is good because it does not get trapped on dead end paths. Best First search combines the advantages of both DFS and BFS into a single method.
A* Search is optimal if heuristic is both consistent and admissible. A* combines the advantages of Best-first Search and Uniform Cost Search: ensure to find the optimized path while increasing the algorithm efficiency using heuristics.

6: Corners Problem: Heuristic

Heuristic is consistent if $h(n) \leq c(n,a,n') + h(n')$ where $h(n)$ is the cost for a node n, $c(n,a,n')$ is the cost to the goal state a through n'. This means that the estimate is always less than or equal to the estimated distance from any neighboring vertex to the goal, plus the cost of reaching that neighbor.
Heuristic is admissible if it never overpredicts the distance to the goal state. This means that $h(n) \leq h^*(n)$, where $h^*(n)$ is the actual cost to reach the goal state.

7: Eating All the Dots

Using BFS can guarantee optimality instead of Manhattan distance, even at a high cost.

8: Suboptimal Search

Write down a descriptive for this example and why you chose this example.

If Pacman needs to eat the nearest dots, we can use BFS. The state will change every time Pacman eats the dots, so we only need to define the target state as if there are dots in the current location, then can reach the target state. With BFS, Pacman can eat all the dots.