

CS182 Assignment 1

Charles Liu

cliu02@g.harvard.edu

9/21/2015

1: The Pac-Man board will show an overlay of the states explored and the order in which they were explored. (Brighter red means earlier exploration.) Is the exploration order the one you would have expected? Does Pac-Man actually go to all the explored squares on his way to the goal? (Question 1 on the Berkeley search project page.)

Yes, with DFS the brighter red looks like a path whereas with BFS the area near the starting point is brighter red. This makes sense since DFS goes on a path until it finds a solution or revisits an old state again whereas BFS is going through all states of the map in order of depth. Pac-Man does not go to all the explored squares on his way to the goal since the goal states visited in the algorithms can be outside of the optimal path - there's no ordering of any costs it's just whatever is next in the container.

2: With regards to question 4 in the implementation of A* as specified by question 4 on the Berkeley search project page: Run the A* agent on openMaze. How do A* and UCS perform with this configuration?

Using my implementations, A* searched 211 nodes whereas UCS search 682. Nearly the entire map was bright red with UCS, but the heuristic made the A* implementation realize that not every move was just an extra 1 cost, there's an associated heuristic cost that would be of impact.

3: Describe a real world problem for which you would want to find an optimal solution. Briefly explain why (3-4 sentences).

If you provide some kind of delivery service for customers, you would want to find the optimal path to deliver, especially if the quality of the product you're delivering deteriorates over time (e.g. food). The cost of unnecessary travel could affect the satisfaction of the customer, both in quality of product and time to wait. The cost function can also be measured in a variety of ways: distance, traffic, quality of roads, etc.

4: Describe a real-world problem for which you would prefer finding a solution quickly, even if it is suboptimal. Briefly explain why (3-4 sentences).

Sometimes calculating an optimum solution is very, very expensive. Suppose you have a huge graph related to Facebook, where a node is a user and an edge is whether or not they're friends. If someone asked you to find someone with the most mutual friends as you, you could go through every single friend of your friend and add it up, or you could look at a few friends that have the most friends and chances are they will have some commonalities.