ISE 308

Sinem Özden
150150202

Q1: My choice is c: public transportation recommendation system

Performance measure: can be changed based on the goal given: finding fastest path, finding shortest path, least transfer path

Environment: is fully observable, semi-dynamic, single agent, episodic, deterministic, discrete, stochastic

Actuator: is the display of map that shows the route

Sensor: are gps signals from public transportation vehicle

Goal based agent would suit this agent best because the performance meause is changed based on the goal set by users preference. A goal based agent also chooses its actions according to their goals.

Q2:

h(n)⇒is a heuristic function that is consistent
n⇒node
n'⇒successor
a⇒action

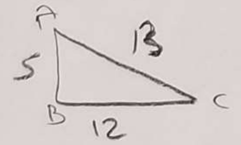
$$\left.\right\} h(n) \leq c(n,a,n') + h(n')$$

Let k(n) is the cheapest path ⇒ We are trying to prove that $\underline{h(n) \leq k(n)}$

Base case: if there aren't any steps between n and goal, then n is the goal. Thus $h(n) = 0 \leq k(n)$

Induction step: Node n is i steps away from goal, so there is a n' successor that generated by action a, From n' to goal best path should have i-1 steps. Thus, $h(n) \leq c(n,a,n') + h(n')$. Thus, by using induction hypothesis $h(n') \leq k(n')$. According to consistency

$h(n) \leq c(n,a,n') + h(n') \leq c(n,a,n') + k(n') = k(n)$



Lets assume that heuristic of points between B and C is smaller than 7 which is 5. We try to go to c from A. The sumation of actual cost from A to B and heuristic from B to C is 5+5=10. which is smaller than 13. So the algorithm will chose the AB-C path instead of A-C which is longer.

Q3: I used the algorithm of “Artificial Intelligence: A Modern Approach” book. The algorithms of BFS DFS and A* similar to the ones in the book.

a) Problem: Eating every peg until there is no vail move left. To eat a peg you must jump on top of the other one.
   Solution: Creating a 7x7 matrix as a board in the program. Then I check all the board if there is any move that I can do. If there any I apply it and save the new form of the board and continue checking it. I defined my acion as a vector of three elements first and second elements are the location of the peg. Last element is to check in which way my action is. Then I tried to move until there is no action left.
b) –
c) –
d) If the heuristic is close to the real cost, then I think A* search won’t be affected and will be ideal. However, BFS run time will get longer because the answer is at the leaf nodes. DFS will also get longer but not as much as DFS.