Assignment Two: Search

Solution

Question One

See: `http://aipython.org/aipython_322_as2/coffee_sol.py`

You can find the answers without MPP by entering the commented out parts at the end of the file in interactive mode of Python.

Heuristic $h1$ uses the Manhattan distance to the goal node.

Heuristic $h2$ uses the Manhattan distance to the goal node if the robot is carrying coffee, otherwise it uses the shortest distance from the current postion to the goal via a coffee shop (the Manhatten distance to the coffee shop plus the Manhatten distance from coffee shop to goal). Note that it needs the shortest distance to ensure it is an underestimate. It uses the sum of the distance to the coffee shop plus the distance from the coffee shop to the goal because the robot has to do both sequentially.

Question Two

`http://aipython.org/aipython_322_as2/grid.py` for a Python implementation of the grid.

With the minus, 21 paths are expanded with or without MPP. Without the minus, 121 paths are expanded with MPP and 520676 without MPP.

The second value in the triple only gets used when the first values are equal. In Chris's example, all (i, j) values where $0 \leq i \leq 10$ and $j \leq i \leq 10$ have the same f-value of 20. The definition of A^* search does not specify what happens when multiple nodes have same f-value. With the minus it implements a (first-in, last-out) stack for these equal values, and so acts like depth-first search, selecting a path that extends the latest path and can go directly to the goal. Without the minus, it implements a (first-in, first-out) queue, and so acts like a breadth-first search exploring multiple paths.

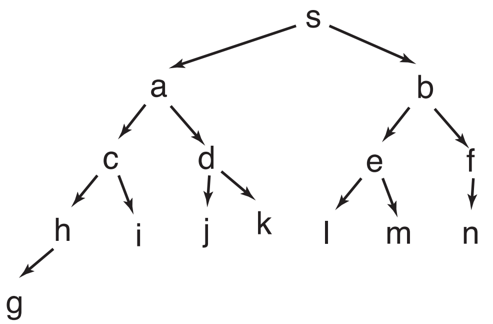
This behaviour occurs whenever there are multiple paths with the optimal f-value. It is better to act like depth-first search in this case than breadth-first search.

Question Three

In all of these graphs, we assume that we are on a plane, with Euclidean distance (straight-line distance) as the arc cost and as the heuristic function. We also assume that the neighbors are ordered from left to right. The start node is s and the goal node is g.

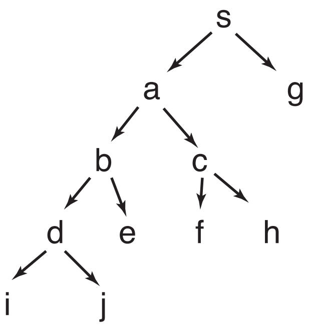
(a) Give a graph where depth-first search is much more efficient (expands fewer nodes) than breadth-first search.

Here breadth-first search expands every node, whereas depth-first search expands five nodes:



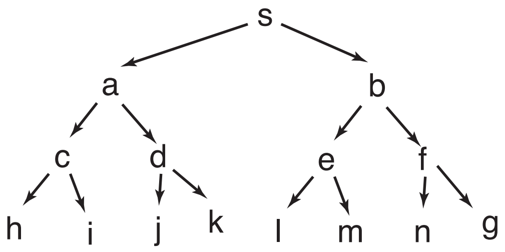
(b) Give a graph where breadth-first search is much better than depth-first search.

Here depth-first search expands every node, whereas breadth-first search expands three nodes:



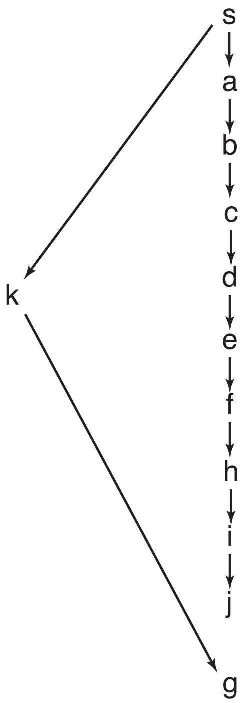
(c) Give a graph where A* search is more efficient than either depth-first search or breadth-first search.

Here depth-first search and breadth-first search expand every node, whereas A* search expands 4 nodes.



(d) Give a graph where depth-first search and breadth-first search are both more efficient than A* search.

Here depth-first search expands three nodes, breadth-first search expands 4, yet A* search expands every node.



Question Four

It should not have taken more than a few hours. Everyone should have worked together to find the solution; if someone was not involved they will not have learned what they were supposed to, and may struggle during the exams. Most of this should have been in understanding the material, not in doing busy work. I hope it was reasonable, and you learned something.