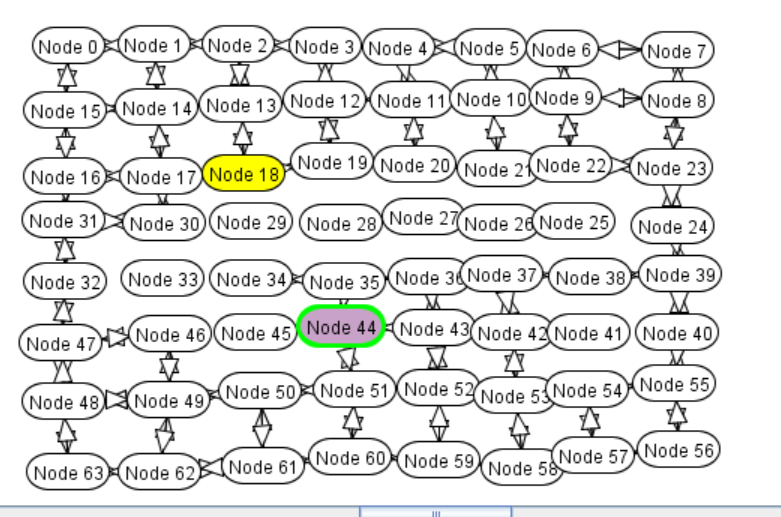
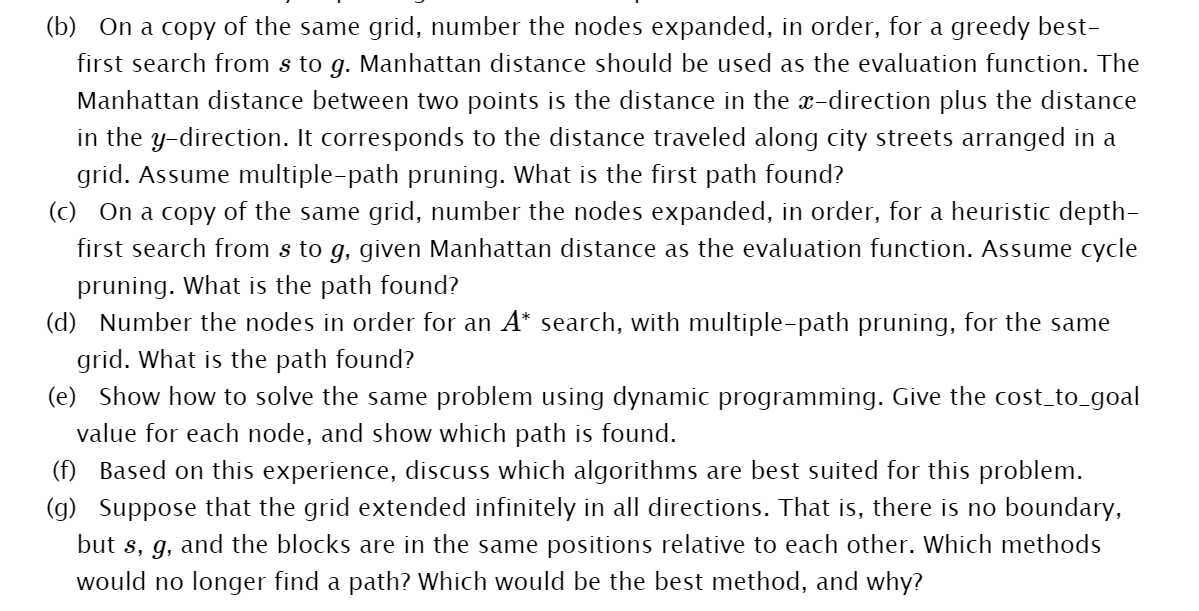
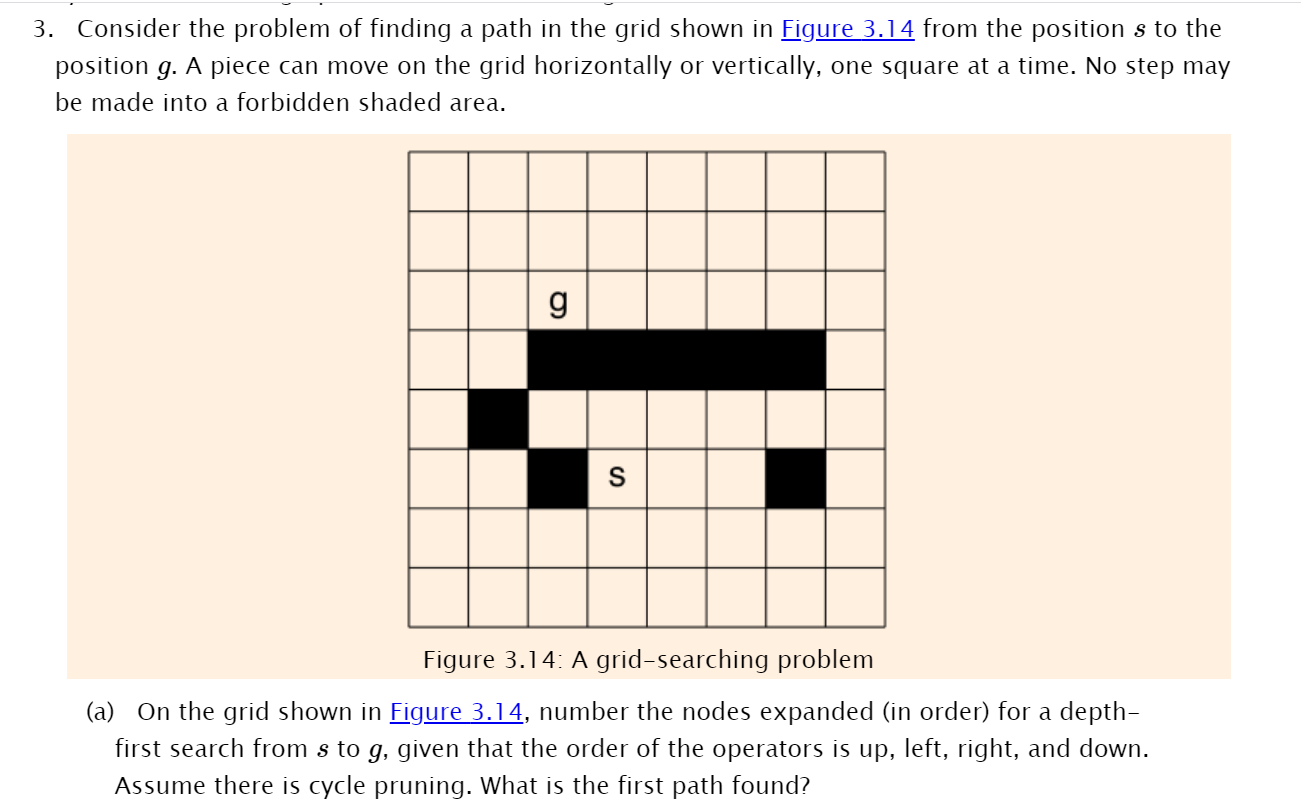
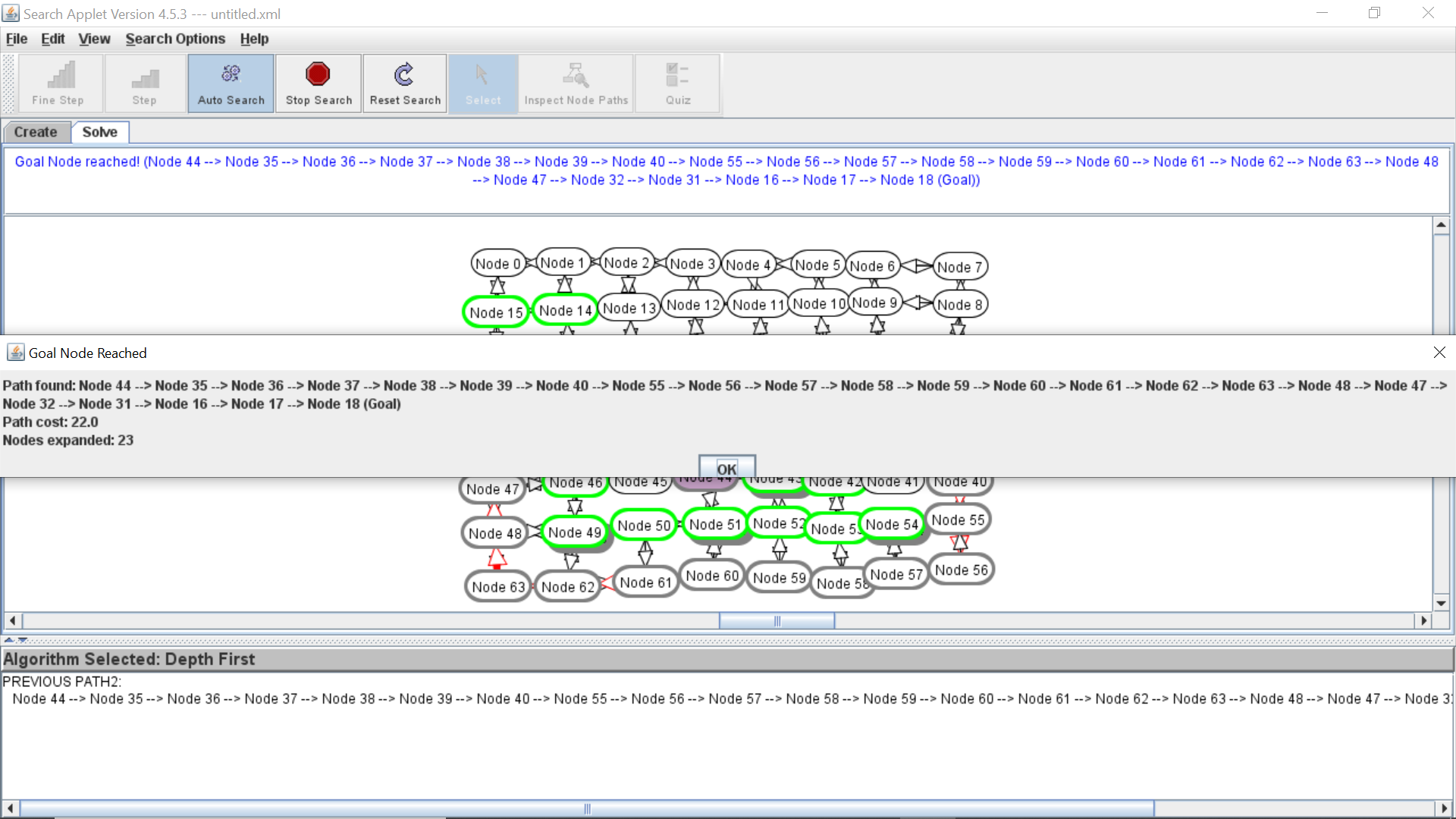
Problem set #2 (20p)

1. (14p) Exercise 3.3 from the book. For parts a-d you should use the AISpace search applet. Note that you need to create an explicit copy of the graph for this problem. You should submit the file with the generated graph with your solution.

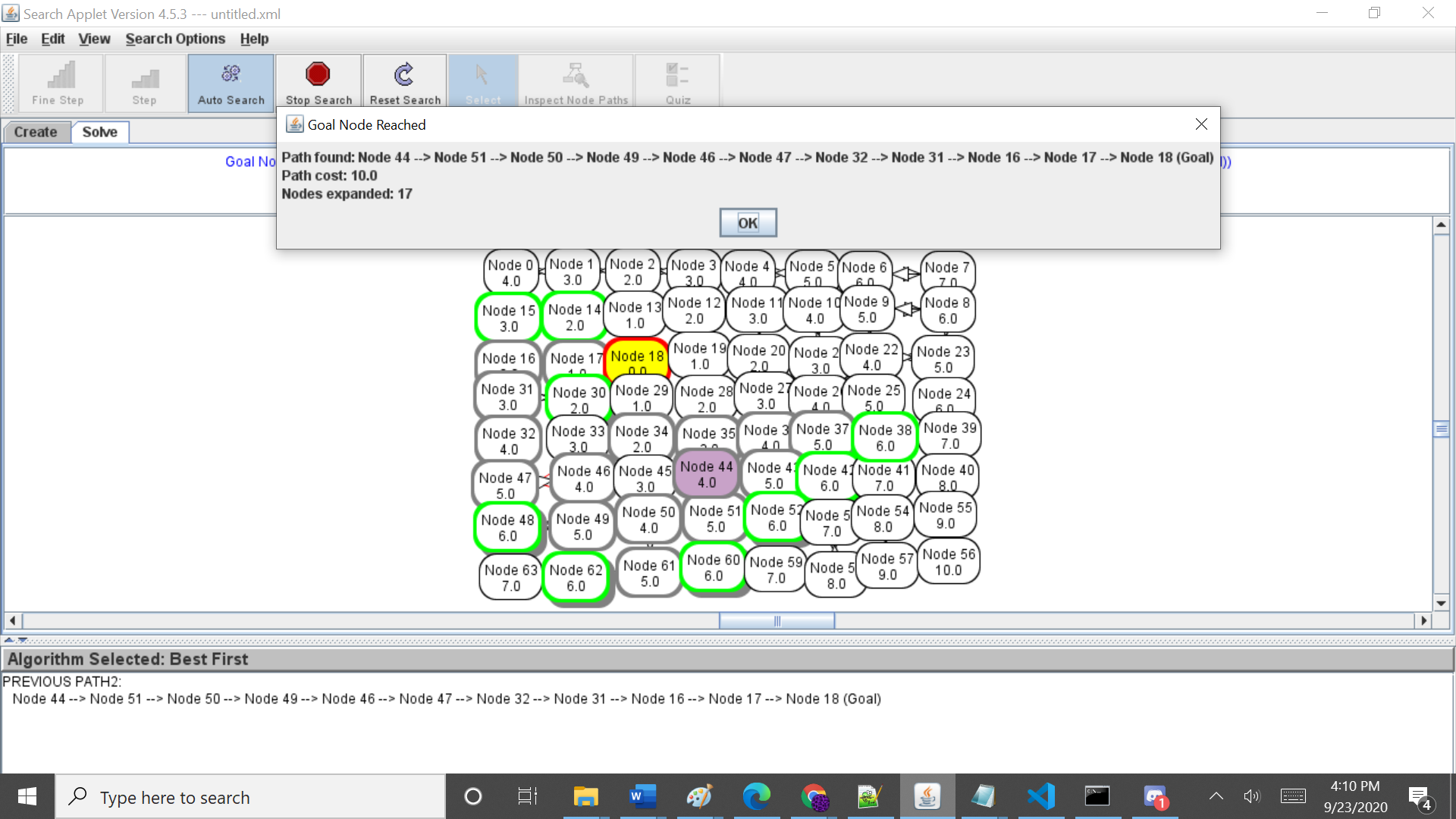


1. Above is a snapshot of my grid.



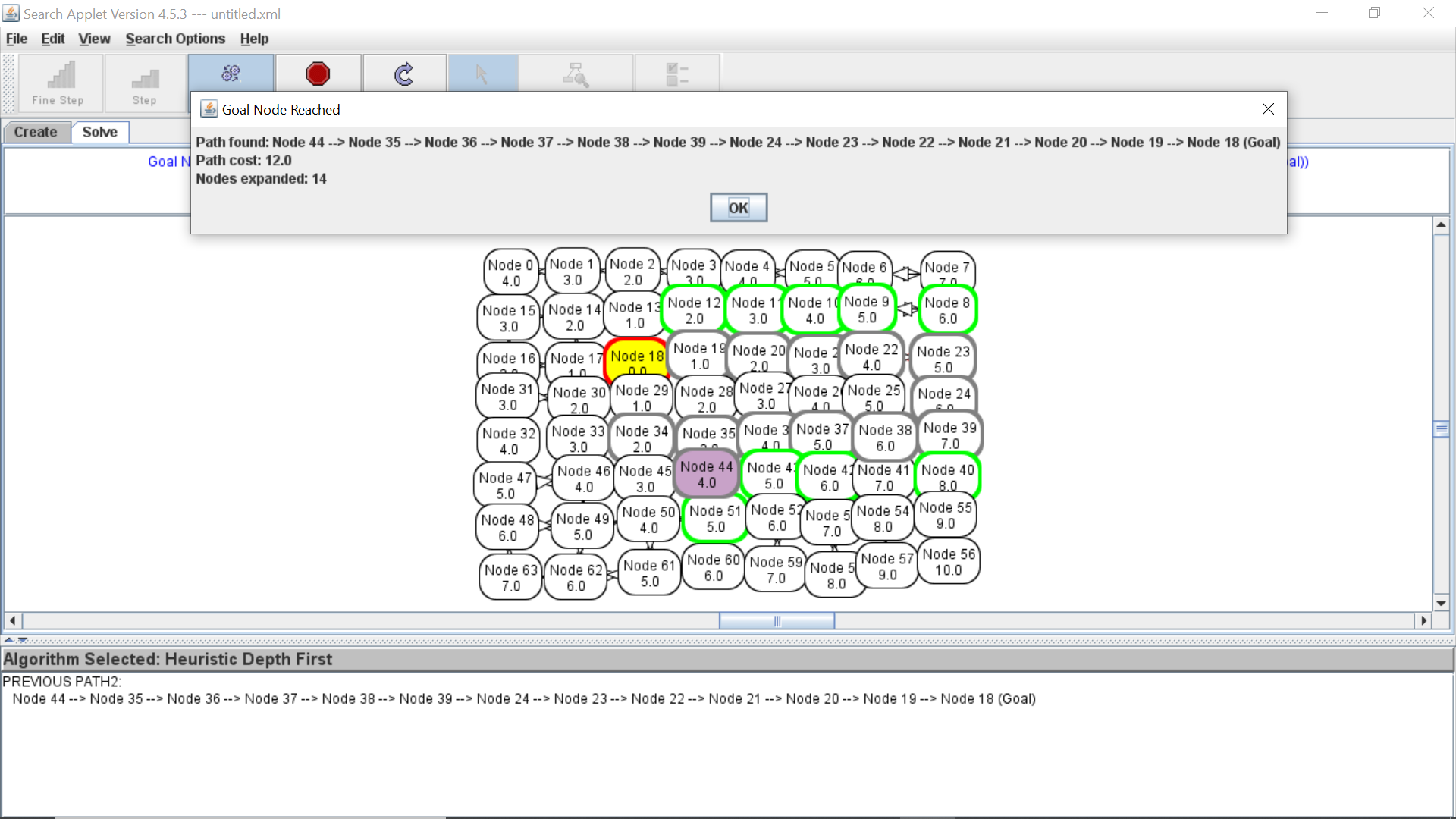
This is the path it found for dfs

b.



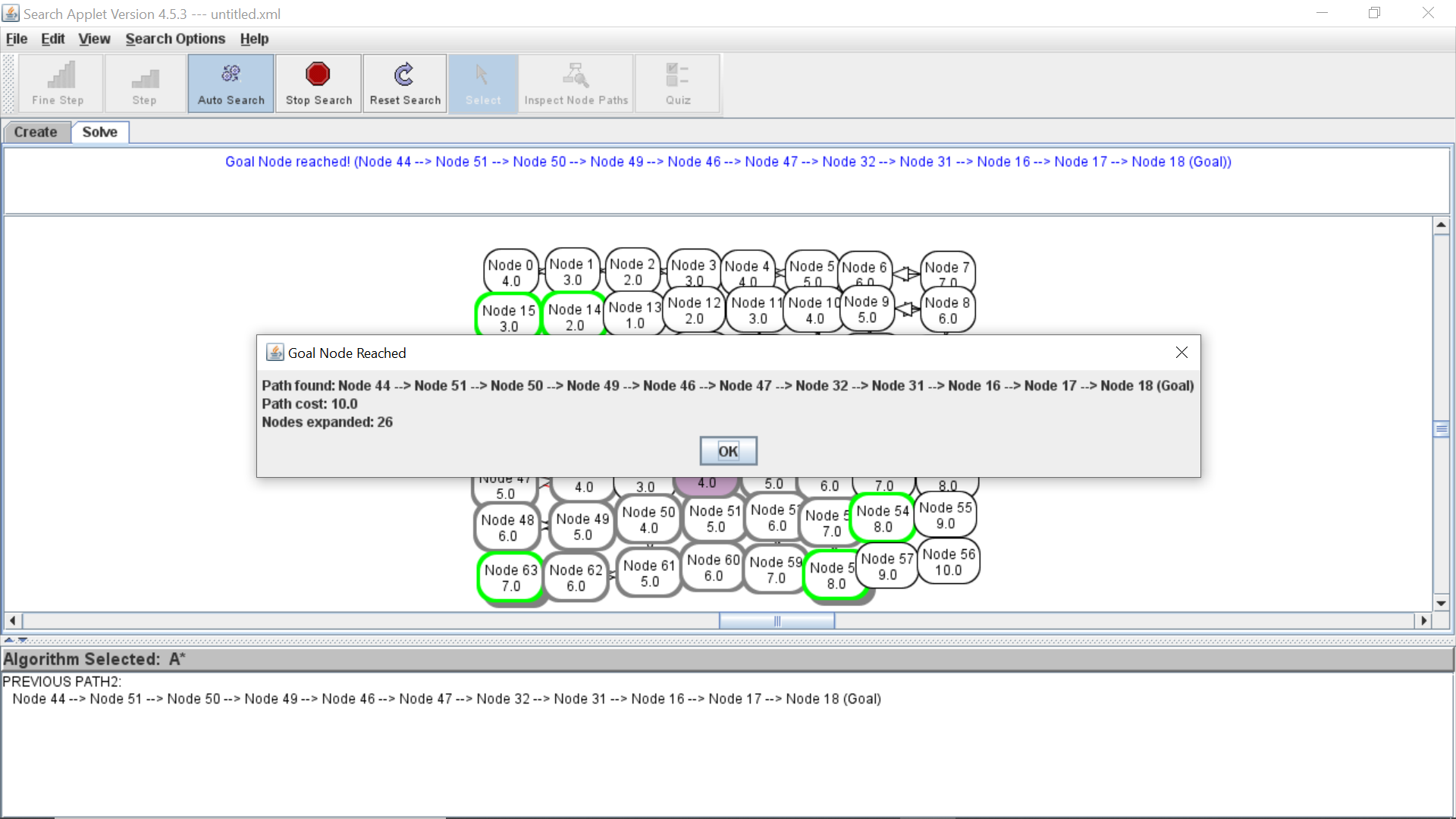
This is for best first search

c.



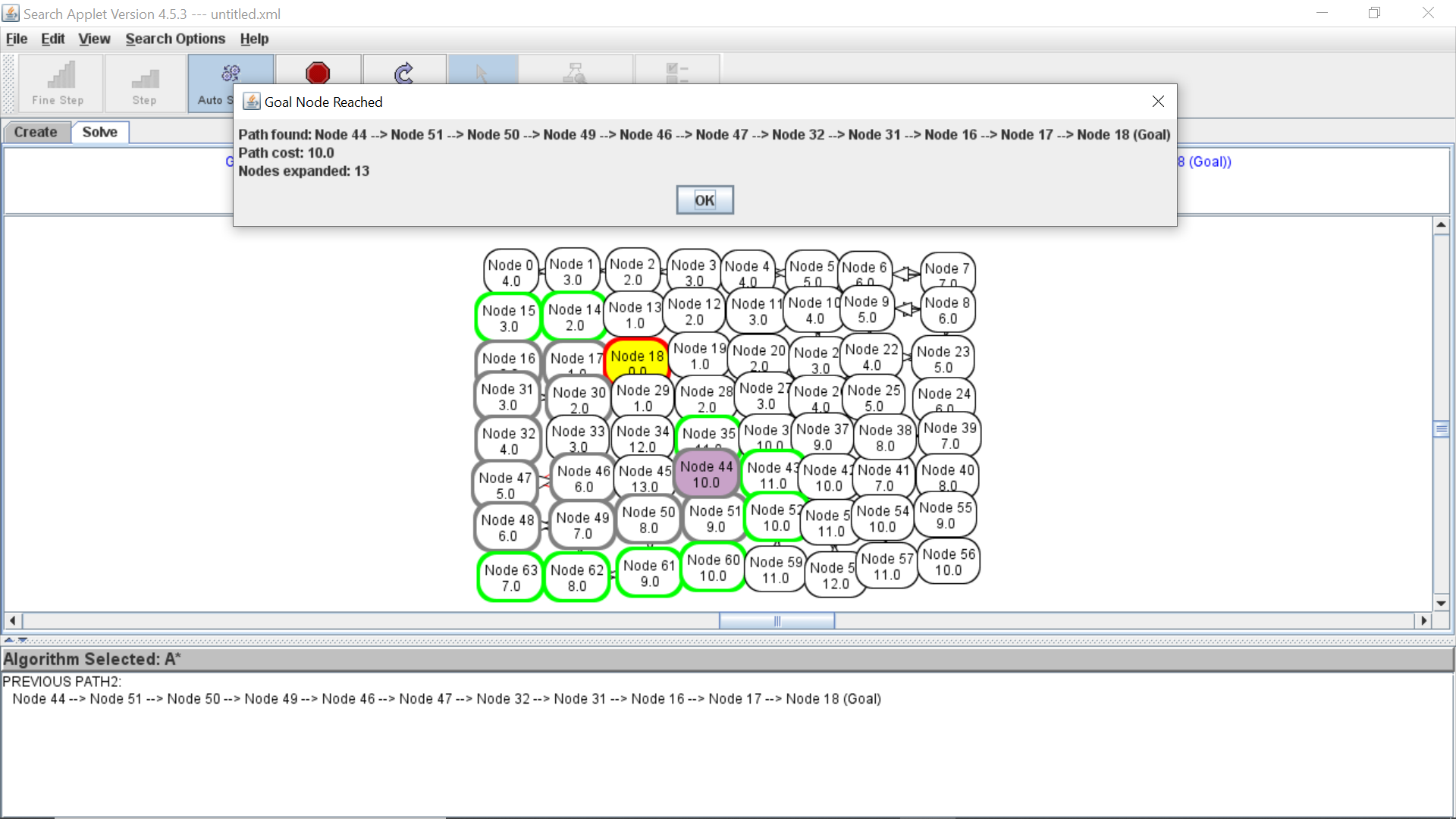
Above is the heuristic depth first search

d.



The results of an A\* search

e.



I changed the heuristic to a better representation of the cost to goal function.

f. Given this expirence A\* search and dynamic programming are the best at finding the best path quickest.

g. If the grid was to be expanded in all directions then depth first would not find a solution. Some kind of limit would need to be placed so it does not got wandering off inifinetly.Dynamic programming would find a solution but continue running. A\* would be the best method because it would find the solution and return it.

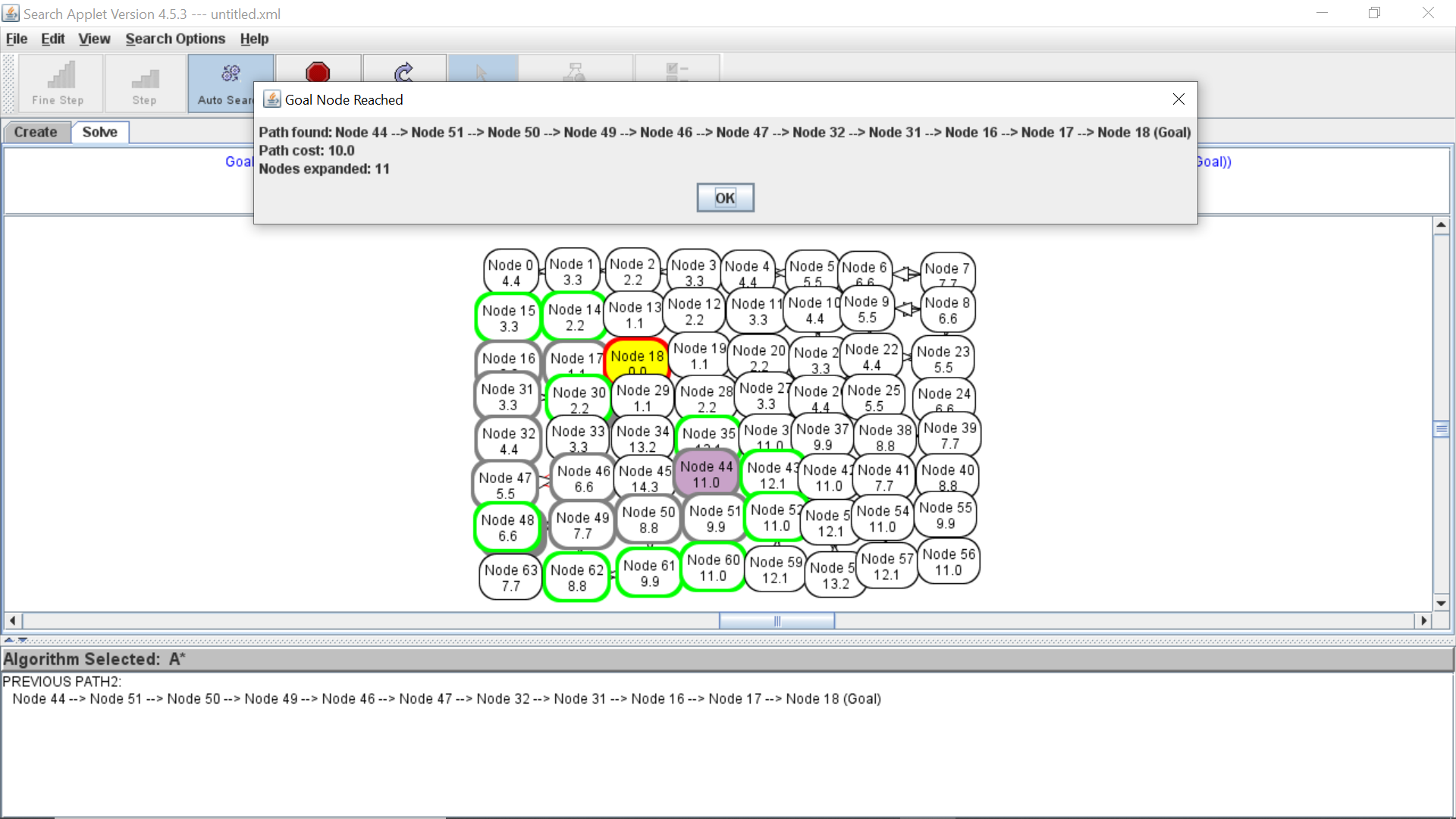
(6p) Exercise 3.7 from the book on the grid from Exercise 3.3. Use AISpace search applet to solve the problem

7.

What happens if the heuristic function is not admissible, but is still nonnegative? What can we say about the path found by A\* if the heuristic function

1. (a)

is less than 1+ϵ1+ϵ times the least-cost path (e.g., is less than 10% greater than the cost of the least-cost path)



This heuristic is not admissible because it is an overestimation. This can cause A\* to sometimes chose the wrong path. In our example the least cost path is ten but if believe it is 10.1 we might overlook a path that is 10.

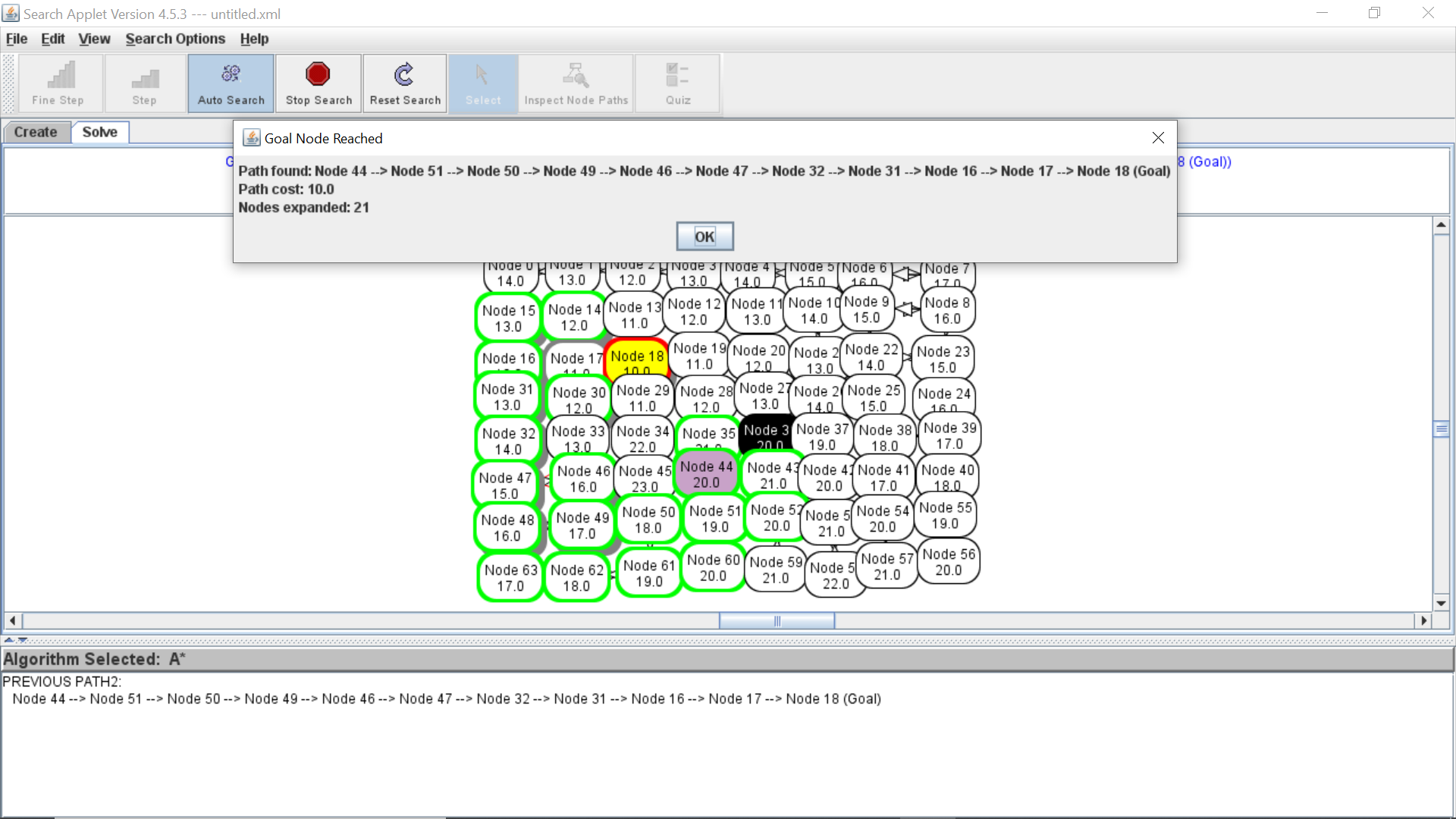
1. (b)

is less than δ more than the least-cost path (e.g., is less than 10 units plus the cost of the optimal path)?

Develop a hypothesis about what would happen and show it empirically or prove your hypothesis. Does it change if multiple-path pruning is in effect or not?

Again, this is going to incorrectly estimate the least cost path. This could again cause the algorithm to choose the incorrect (not best) path as a solution.

Removing multiple path pruning causes more nodes to be explored.



Does loosening the heuristic in either of these ways improve efficiency? Try A\* search where the heuristic is multiplied by a factor 1+ϵ, or where a cost δ is added to the heuristic, for a number of graphs. Compare these on the time taken (or the number of nodes expanded) and the cost of the solution found for a number of values of ϵ or δ.

No it does not improve efficiency to loosen these heuristics. A heuristic function should estimate the true cost correctly to quickly reach the correct solution.