William Burkes

CSC412

Assignment 2

3rd Edition

**3.23**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Action | Open Nodes | f(n) | g(n) | h(n) |
| 1  2  3  4  5  6  7  8 | Initial State  Expand Lugoj  Add Timisoara  Add Mehadia  Expand Mehadia  Add Drobeta  Expand Drobeta  Add Craiova  Expand Craiova  Add Rimmicu Vilcea  Add Pitesti  Expand Timisoara  Add Arad  Expand Pitesti  Add Bucharest  Expand Bucharest  At Goal State | (L,244)  (T,440)  (T,440),(M,301)  (T,440),(D,387)  (T,440),(C,425)  (T,440),(R,604)  (T,440),(R,604),(P,503)  (R,604),(P,503),(A,595)  (R,604),(A,595),(B,504)  (R,604),(A,595) | 244  440  310  387  425  604  503  595  504  - | 0  111  70  145  265  411  403  229  504  - | 244  329  241  242  160  193  100  366  0  - |

**3.25**

Complete whenever 0 ≤ w ≤ 2.

w = 0 gives f(n) = 2g(n). This is the same behavior as uniform-cost search

w = 1 gives the A\* search

w = 2 gives f(n) = 2h(n). This is the same as greedy best-first search

behaves as A\* with a heuristic

**3.21**

a) When all steps cost equal, g(n) ∞ depth(n) so uniform-cost search reproduces breadth-first search

b) Breadth first search is best first search with f(n) = depth(n)

Depth first is best first with f(n) = -depth(n)

Uniform cost is best first with f(n) = g(n)

c) Uniform cost search is A\* with h(n) = 0

**3.30**

a) The TSP problem is to find the shortest path through the cities that form a closed loop. MST finds the shortest path but doesn’t need to be a closed loop. MST is admissible because it is always shorter than or equal to a closed loop.

b) Straight line distance back to the start city isn’t a good heuristic. It doesn’t do well when there are a lot of cities. In the late stages of a search when there aren’t many cities left it is more viable. MST dominates straight line distance because it includes the goal node and the current node must either be the straight line between them or include two or more lines that add to more