

HOME WORK-3

Venkata Krishnamohan Sunkara

65528957

2.

- a. This problem can be represented as a graph where the nodes correspond to regions on the map and the edges are represented when two regions are adjacent to each other.

Initial State: A graph where all the nodes (regions) are uncolored.

Successor function: Assigning a color to a node (region).

Goal test: All the nodes (regions) are colored with less than 4 different colors and no two adjacent nodes have same color.

Step cost: Is going to be 1 as only one color is assigned to each node in a step.

Path cost: Is going to be the number of color assignments.

- b. Initial state: A state where monkey, crates and bananas are somewhere in the room, that is some random arrangement.

Successor Function: The monkey can move up, down on the crate or move left, right in the room or eat the bananas. The crates can also be moved left, right or can be stacked upon each other.

Goal test: Did the monkey get all the bananas?

Step cost: Every movement is counted and assumed non-negative.

Path cost: Total sum of all the movements done.

- d. Initial state: A state where all the jugs are empty.

Successor function: Fill or empty a jug. If two jugs have current values x , y then pouring water from y to x changes the quantity in jug x to minimum of $x+y$ and the capacity of jug x and decrements jug y by the amount gained by first jug.

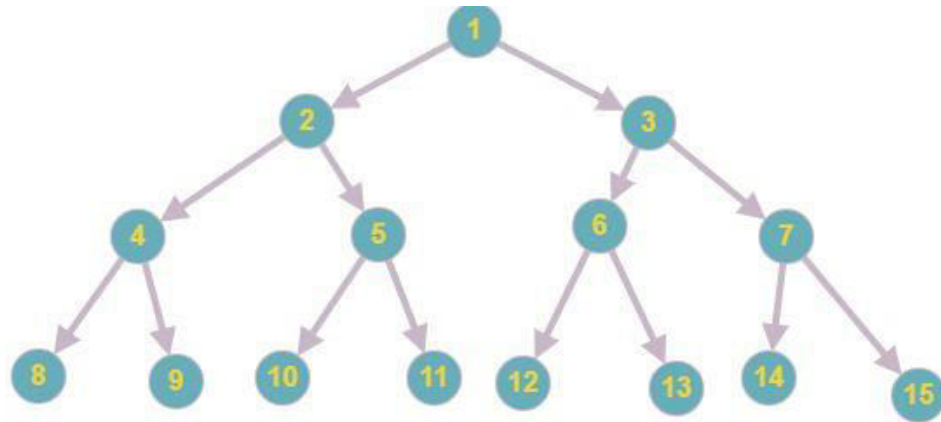
Goal test: Does any one of the jug contain exactly one gallon.

Step cost: Assumed Non-negative.

Path cost: The total sum of all the step costs of actions taken.

3.

a. State space for 1 to 15



b.

Breadth First Search: 1 2 3 4 5 6 7 8 9 10 11

Depth Limited Search: 1 2 4 8 9 5 10 11

Iterative deepening search: 1; 1 2 3; 1 2 4 5 3 6 7; 1 2 4 8 9 5 10 11.

c. The bidirectional search is very useful in this scenario because the predecessor of a node 'n' is a single node and can be obtained by the floor of $(n/2)$. The branching factor in forward direction is 2 and the branching factor in backward direction is 1.

d. Yes, by starting at the goal and proceeding backwards by visiting nodes obtained by floor of $(n/2)$ and continuing until we reach node '1'.

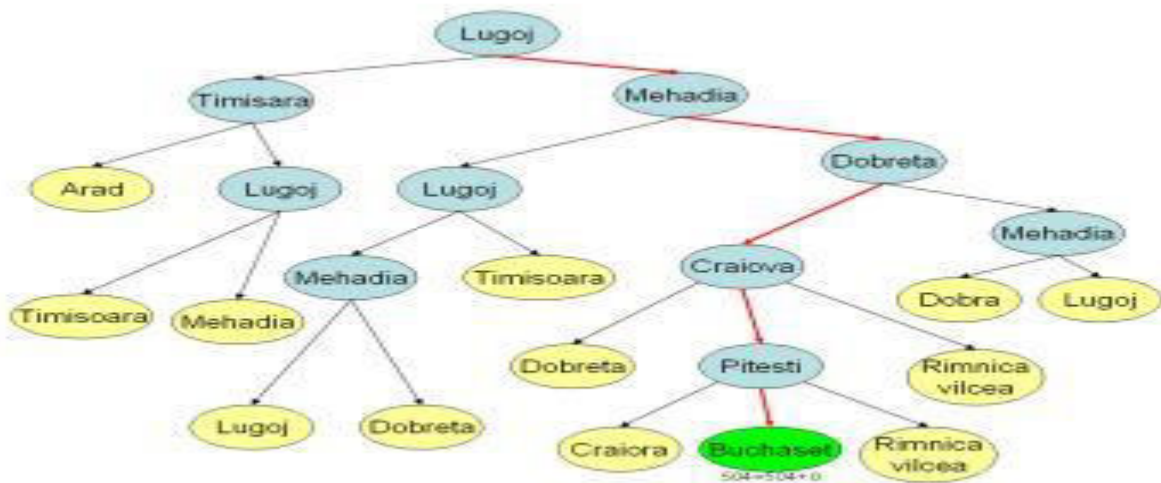
e. If we obtain the goal state 'n', then we can traverse backwards until we reach node '1'. If $(n/2)$ is equal to an integer then push the action as 'left' into a stack, otherwise push the action as 'right' into a stack and continue until node '1' is reached. Now pop all the elements in the stack which gives us a solution to this problem.

Another method to obtain solution is represent the goal state number in binary. Leaving the first bit, if the bit is 0 then assign left action otherwise assign right action. Now the bits apart from 1st bit tells us the action sequence required to obtain a solution.

4.

i. Suppose, if we run a greedy search algorithm with $h(n) = -g(n)$, then the greedy search emulates a Depth first search because the greedy search chooses the node from the fringe whose $h(n)$ value is less, which is the one with highest $g(n)$. So, if a node is expanded then its $g(n)$ value increases which means the $h(n)$ value decreases and that node is expanded next and this process continues in a depth first fashion.

- ii. Suppose, if we run a greedy search algorithm with $h(n) = g(n)$, then the greedy search emulates a Uniform cost search because the greedy search chooses the node from the fringe whose $h(n)$ value is minimum which is the one with lowest $g(n)$. This is Uniform cost search.
5.
 - a. When all the step costs are equal, then uniform cost search expands all the nodes in a breadth - first fashion. Here $g(n)$ is proportional to the depth of search tree.
 - b. Depth-first search is a special case of best-first tree search, if the evaluation function $f(n)$ is equal to $-(\text{depth}(n))$ then the deeper nodes from the fringe are expanded first and it continues in a depth first manner.
 - c. Uniform cost search is a special case of A* search, if the evaluation function $f(n)$ of A* search is equal to $g(n)$ and the heuristic function is ' $h(n) = 0$ ' then the A* search condenses to Uniform cost search as $f(n)=g(n)$.
- 6.



The search tree for tracing the route from lugoj to Bucharest by A* search is depicted above.

The sequence of nodes that the algorithm will consider are:

1. Lugoj: $g(n) = 0$; $h(n) = 244$; $f(n) = 244$
2. Mehadia: $g(n) = 70$; $h(n) = 241$; $f(n) = 311$
3. Timisoara: $g(n) = 111$; $h(n) = 329$; $f(n) = 440$
Now expand the mehadia node.
4. Lugoj (1): $g(n) = 140$; $h(n) = 244$; $f(n) = 384$
5. Dobreta: $g(n) = 145$; $h(n) = 242$; $f(n) = 387$
Now expand the Lugoj (1) node.
6. Mehadia (1): $g(n) = 210$; $h(n) = 241$; $f(n) = 451$
7. Timisoara (1): $g(n) = 251$; $h(n) = 329$; $f(n) = 580$
Now expand the Dobreta node.

8. Mehadia (2): $g(n) = 220$; $h(n) = 241$; $f(n) = 461$

9. Craiova: $g(n) = 265$; $h(n) = 160$; $f(n) = 425$

Now expand the Craiova node.

10. Dobreta (1): $g(n) = 385$; $h(n) = 242$; $f(n) = 627$

11. Rimnicu Vilcea: $g(n) = 411$; $h(n) = 193$; $f(n) = 504$

12. Pitesti: $g(n) = 403$; $h(n) = 100$; $f(n) = 503$

Now expand Timisoara node.

13. Lugoj (2): $g(n) = 222$; $h(n) = 244$; $f(n) = 466$

14. Arad: $g(n) = 229$; $h(n) = 366$; $f(n) = 595$

Now expand Mehadia (1)

15. Lugoj (3): $g(n) = 280$; $h(n) = 244$; $f(n) = 524$

16. Dobreta (2): $g(n) = 285$; $h(n) = 242$; $f(n) = 527$

Now expand Mehadia (2)

17. Lugoj (4): $g(n) = 290$; $h(n) = 244$; $f(n) = 534$

18. Dobreta (3): $g(n) = 295$; $h(n) = 242$; $f(n) = 487$

Now expand Pitesti node

19. Craiova (1): $g(n) = 541$; $h(n) = 160$; $f(n) = 601$

20. Rimnicu Vilcea (1): $g(n) = 500$; $h(n) = 193$; $f(n) = 693$

21. Bucharest: $g(n) = 504$; $h(n) = 0$; $f(n) = 504$

Now expand Lugoj (2) node

22. Mehadia (3): $g(n) = 292$; $h(n) = 241$; $f(n) = 533$

23. Timisoara (2): $g(n) = 333$; $h(n) = 329$; $f(n) = 662$

Now the lowest $f(n)$ is by Bucharest node and before expanding it we check whether it is a goal. So, we can stop the search here as we reached the goal node.

The path to goal node consists of Lugoj, Mehadia, Dobreta, Craiova, Pitesti and The Bucharest nodes.