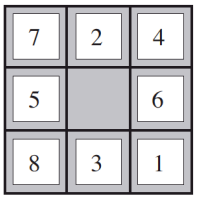
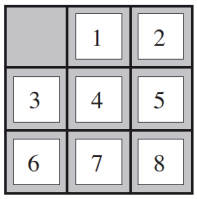
1. Uniform-cost search expands the node n with the lowest path cost g(n). What is g(n)?
2. What data structures do BFS, Uniform-cost search, and DFS use?
3. What is the difference between Uniform-cost Search and Greedy Best-first Search?
4. Discuss the two commonly used heuristics for the 8-puzzle game. Use the following diagrams for your calculations.

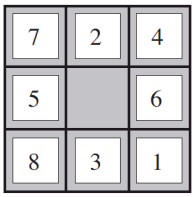


Start State

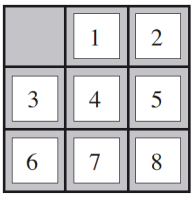


Goal State

5. Discuss the "Manhattan distance" heuristic for the 8-puzzle game. Use the following diagrams for your calculations.

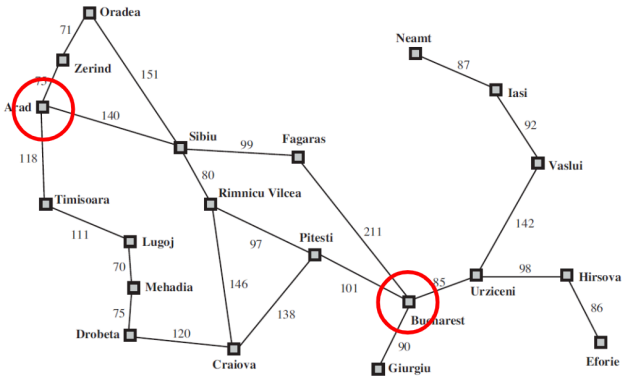


Start State



Goal State

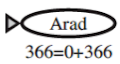
6. What are the advantages of DFS over BFS? Explain in the context of space complexity.
7. Write the DFS algorithm.
8. If g(n) is the cost to reach the node, and h(n) is the cost to get from the node to the goal, what is the difference between Uniform Cost search and A* search?
9. Given the following map (with the path costs shown) and the table with the heuristic function, we are interested in the stages of the A* search algorithm. Arad is the starting state and Bucharest is the goal state. As the A* search is executed, after Sibiu, the next node to be expanded is Rimnicu Vilcea. With calculations explain why.



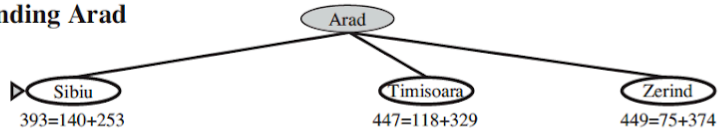
Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

Values of h_{SLD}—straight-line distances to Bucharest.

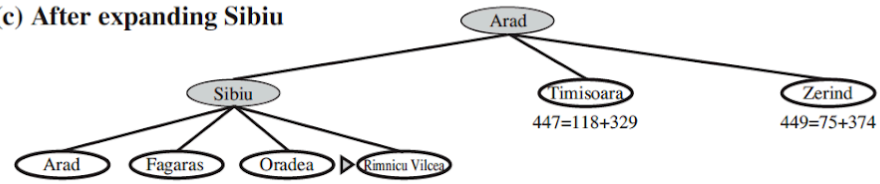
(a) The initial state



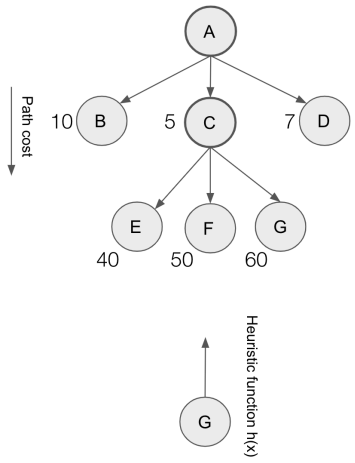
(b) After expanding Arad



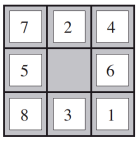
(c) After expanding Sibiu



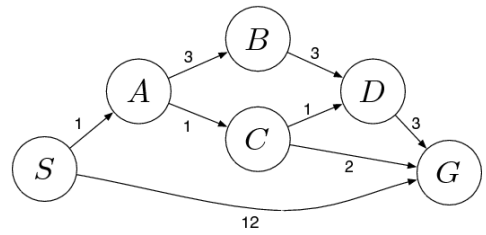
10. Given node A, having child nodes B, C, D with associated costs of (10, 5, 7). After expanding C, we see nodes E, F, G with costs of (40, 50, 60). Which node will be chosen by Uniform-cost search and Greedy best-first search? Explain with the following diagram as reference.



11. For the 8-puzzle game, what actions are possible in a given random state?



12. What will be the contents of the priority queue, as the UCS algorithm proceeds on the following graph? Show the contents of the priority queue, step-by-step.



function UNIFORM-COST-SEARCH(*problem*) **returns** a solution, or failure

node ← a node with STATE = *problem*.INITIAL-STATE, PATH-COST = 0
frontier ← a priority queue ordered by PATH-COST, with *node* as the only element
explored ← an empty set
loop do
  if EMPTY?(*frontier*) **then return** failure
  node ← POP(*frontier*) /* chooses the lowest-cost node in *frontier* */
  if *problem*.GOAL-TEST(*node*.STATE) **then return** SOLUTION(*node*)
  add *node*.STATE to *explored*
  // Print the contents of the Priority Queue (frontier)
  for each *action* **in** *problem*.ACTIONS(*node*.STATE) **do**
    child ← CHILD-NODE(*problem*, *node*, *action*)
    if *child*.STATE is not in *explored* or *frontier* **then**
      frontier ← INSERT(*child*, *frontier*)
    else if *child*.STATE is in *frontier* with higher PATH-COST **then**
      replace that *frontier* node with *child*