Artificial Intelligence and Expert System

Mid Semester Assignment

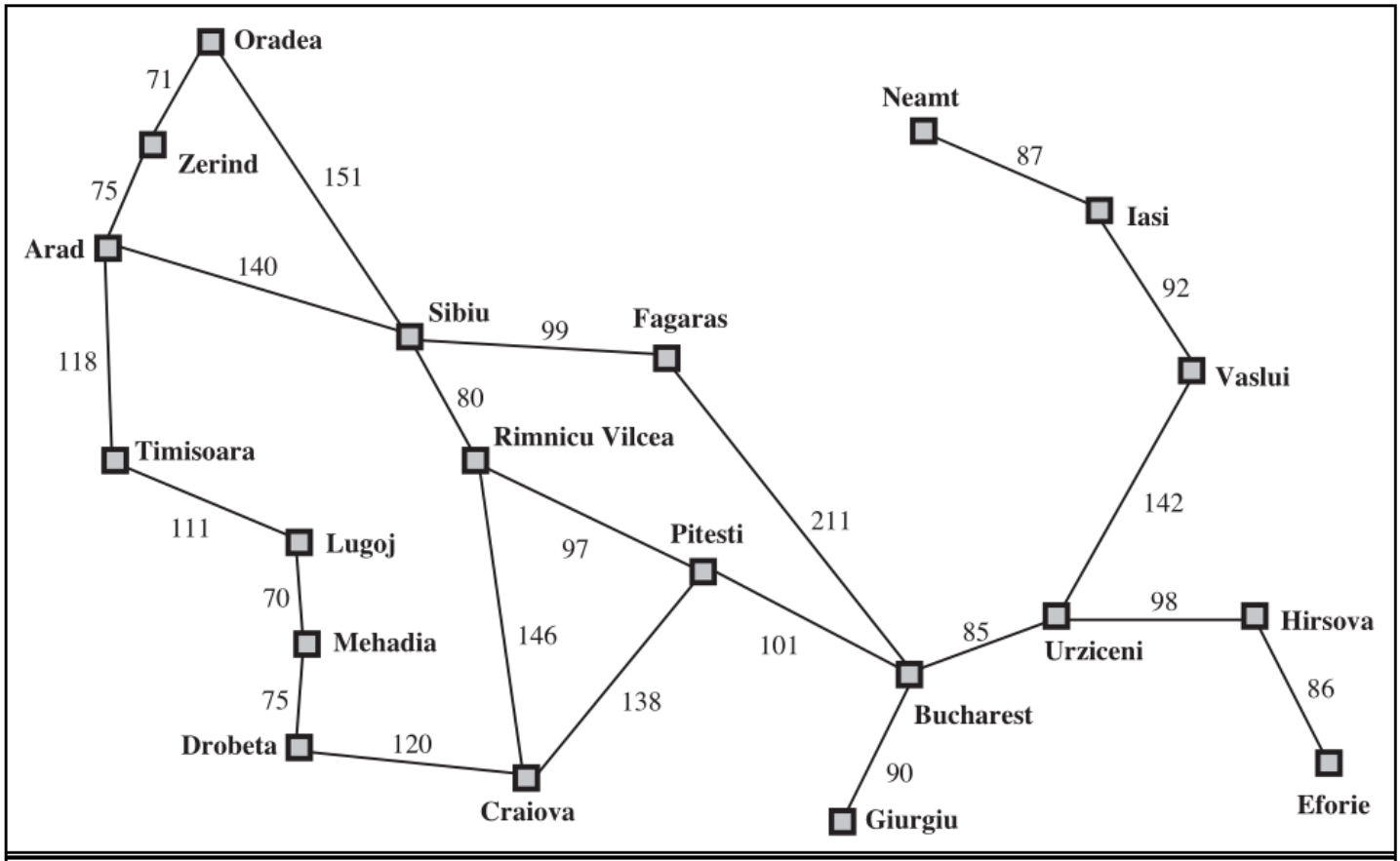

Figure 1 A simplified road map of part of Romania.

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

Figure 2 Values of h_{SLD}—straight-line distances to Bucharest.

Question:

Find a path from ***Zerind*** to ***Bucharest*** using the four different algorithms described below:

1. Breadth First Search
2. Uniform Cost Search.
3. Greedy Best First Search
4. A* Search
5. Compare the result of these four algorithms in perspective of **time** and **optimality**.

Instruction:

1. You have to answer in the space given below.
2. Write your name and id in the given space below.
3. Save this file to pdf and upload in your portal account.
4. Submission deadline: 21/11/20

Name: Tawhid Khondakar	**ID: 18-36611-1**	**Section: E**

Answer 1: Breadth First Search

Step	Frontier	Explored Set	Expand
1	{ (Zerind,0) }	{ }	Zerind
2	{ (Oradea,71), (Arad,75) }	{ Zerind }	Oradea
3	{ (Arad,75), (Sibiu,222) }	{ Zerind, Oradea }	Arad
4	{ (Sibiu,222), (Timisoara,193), (Sibiu,215) }	{ Zerind, Oradea, Arad }	Sibiu
5	{ (Timisoara,193), (Fagaras,321), (Rimnicu Vilcea,302) }	{ Zerind, Oradea, Arad, Sibiu }	Timisoara
6	{ (Fagaras,321), (Rimnicu Vilcea,302), (Lugoj,304) }	{ Zerind, Oradea, Arad, Sibiu, Timisoara }	Fagaras
7	{ (Rimnicu Vilcea,302), (Lugoj,304), (Bucharest,532) }	{ Zerind, Oradea, Arad, Sibiu, Timisoara, Fagaras }	Rimnicu Vilcea
8	{ (Lugoj,304), (Bucharest,532), (Craiova,448), (Pitesti,399) }	{ Zerind, Oradea, Arad, Sibiu, Timisoara, Fagaras, Rimnicu Vilcea }	Lugoj
9	{ (Bucharest,532), (Craiova,448), (Pitesti,399), (Mehadia,374) }	{ Zerind, Oradea, Arad, Sibiu, Timisoara, Fagaras, Rimnicu Vilcea, Lugoj }	Bucharest
10			
11			
12			
13			
14			
15			

Answer 2: Uniform-Cost Search

Step	Frontier	Explored Set	Expand
1	{ (Zerind,0) }	{ }	Zerind
2	{ (Oradea,71), (Arad,75) }	{ Zerind }	Oradea
3	{ (Arad,75), (Sibiu,222) }	{ Zerind, Oradea }	Arad
4	{ (Sibiu,222), (Timisoara,193), (Sibiu,215) }	{ Zerind, Oradea, Arad }	Timisoara
5	{ (Sibiu,222), (Sibiu,215), (Lugoj,304) }	{ Zerind, Oradea, Arad, Timisoara }	Sibiu
6	{ (Lugoj,304), (Fagaras,314), (Rimnicu Vilcea,295) }	{ Zerind, Oradea, Arad, Timisoara, Sibiu }	Rimnicu Vilcea
7	{ (Lugoj,304), (Fagaras,314), (Craiova,441), (Pitesti,392) }	{ Zerind, Oradea, Arad, Timisoara, Sibiu, Rimnicu Vilcea }	Lugoj
8	{ (Fagaras,314), (Craiova,441), (Pitesti,392), (Mehadia,374) }	{ Zerind, Oradea, Arad, Timisoara, Sibiu, Rimnicu Vilcea, Lugoj }	Fagaras
9	{ (Craiova,441), (Pitesti,392), (Mehadia,374), (Bucharest,525) }	{ Zerind, Oradea, Arad, Timisoara, Sibiu, Rimnicu Vilcea, Lugoj, Fagaras }	Mehadia
10	{ (Craiova,441), (Pitesti,392), (Bucharest,525), (Drobeta,449) }	{ Zerind, Oradea, Arad, Timisoara, Sibiu, Rimnicu Vilcea, Lugoj, Fagaras, Mehadia }	Pitesti
11	{ (Craiova,441), (Bucharest,525), (Drobeta,449), (Craiova,530), (Bucharest,493) }	{ Zerind, Oradea, Arad, Timisoara, Sibiu, Rimnicu Vilcea, Lugoj, Fagaras, Mehadia, Pitesti }	Craiova
12	{ (Bucharest,525), (Drobeta,449), (Bucharest,493), (Drobeta,561) }	{ Zerind, Oradea, Arad, Timisoara, Sibiu, Rimnicu Vilcea, Lugoj, Fagaras, Mehadia, Pitesti, Craiova }	Drobeta
13	{ (Bucharest,525), (Bucharest,493) }	{ Zerind, Oradea, Arad, Timisoara, Sibiu, Rimnicu Vilcea, Lugoj, Fagaras, Mehadia, Pitesti, Craiova, Drobeta }	Bucharest
14			
15			

Answer 3: Greedy Best First Search

Step	Frontier	Explored Set	Expand
1	{ (Zerind,0) }	{ }	Zerind
2	{ (Oradea,71), (Arad,75) }	{ Zerind }	Arad
3	{ (Oradea,71), (Timisoara,193), (Sibiu,215) }	{ Zerind, Arad }	Sibiu
4	{ (Oradea,71), (Timisoara,193), (Oradea,366), (Fagaras,314), (Rimnicu Vilcea,295) }	{ Zerind, Arad, Sibiu }	Fagaras

5	{ (Oradea,71), (Timisoara,193), (Oradea,366), (Rimnicu Vilcea,295), (Bucharest,525) }	{ Zerind, Arad, Sibiu, Fagaras }	Bucharest
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

Answer 4: A* Search

Step	Frontier	Explored Set	Expand
1	{ (Zerind,0), }	{ }	Zerind
2	{ (Oradea,71), (Arad,75), }	{ Zerind, }	Arad
3	{ (Oradea,71), (Timisoara,193), (Sibiu,215), }	{ Zerind, Arad, }	Oradea
4	{ (Timisoara,193), (Sibiu,215), (Sibiu,222), }	{ Zerind, Arad, Oradea, }	Sibiu
5	{ (Timisoara,193), (Fagaras,314), (Rimnicu Vilcea,295), }	{ Zerind, Arad, Oradea, Sibiu, }	Rimnicu Vilcea
6	{ (Timisoara,193), (Fagaras,314), (Craiova,441), (Pitesti,392), }	{ Zerind, Arad, Oradea, Sibiu, Rimnicu Vilcea, }	Fagaras
7	{ (Timisoara,193), (Craiova,441), (Pitesti,392), (Bucharest,525), }	{ Zerind, Arad, Oradea, Sibiu, Rimnicu Vilcea, Fagaras, }	Pitesti
8	{ (Timisoara,193), (Craiova,441), (Bucharest,525), (Craiova,530), (Bucharest,493), }	{ Zerind, Arad, Oradea, Sibiu, Rimnicu Vilcea, Fagaras, Pitesti, }	Bucharest
9			
10			
11			
12			
13			
14			
15			

Answer 5: Compare the result of these four algorithms in perspective of time and optimality.

Time and optimality are the two co-fundamental parameters to justify the efficiency of an algorithm. The following problem solutions have been ranked with respect to time and optimality.

Time

1. Greedy Best First Search: Uses heuristics functions to determine the closest path to the destination to reduce time. It took 5 steps to solve this problem.
2. A^* Search: Uses both heuristics function as well as path cost to maintain both optimality and time. Thus it may require more time in some cases. It took total 8 steps to solve this problem.
3. Breadth First Search: Traverse the graph starting from the lowest level to the highest level. It gives better results if the goal state is near to the initial state. For the following problem, It took total 9 steps.
4. Uniform Cost Search: The only concern in this algorithm is making path cost to reach the goal state as minimum as possible. It took total 13 steps for this problem.

Optimality

Optimality depends on the net cost for the associative path that is derived from the algorithm. The fewer paths cost, the more optimal an algorithm is. Here the optimality of these algorithms has given below:

1. A^* search, Uniform Cost Search: Both of the algorithm gives 493 units as the path cost. But here A^* took fewer steps to derive that path.
2. Greedy Best First Search: It takes total 525 units as the path cost.
3. Breadth First Search: This algorithm gives 532 as the path cost.