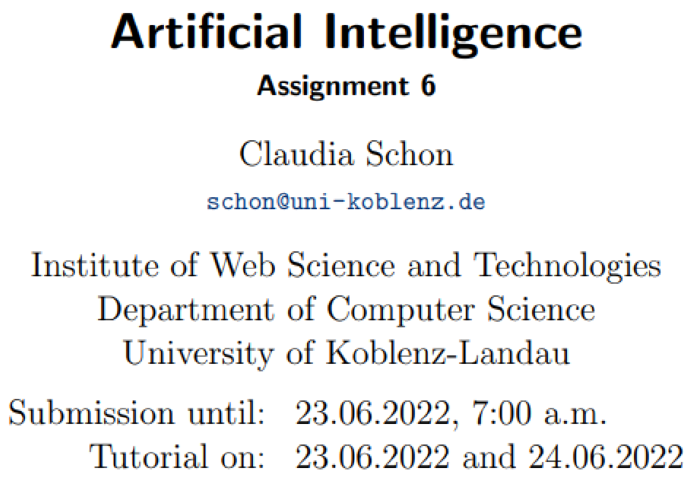


Artificial Intelligence

Assignment 6

Claudia Schon

schon@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Submission until: 23.06.2022, 7:00 a.m.
Tutorial on: 23.06.2022 and 24.06.2022

GROUP HOLLERITH - SOLUTION

Group Members:

1. **Saborni Shernaj Binte Elahi (220202426) – (saborni@uni-koblenz.de)**
2. **M Rashedul Hasnat (220202415) – (rhasnat@uni-koblenz.de)**
3. **Basitur Rahman Chowdhury (218100976) – (bchowdhury@uni-koblenz.de)**
4. **Kamrun Nahar(220202410) – (nahar@uni-koblenz.de)**

1 Uninformed Search (24 Points):

Figure 1 shows a complete search tree of a search problem. The start node is node A, all target nodes are marked with a square.

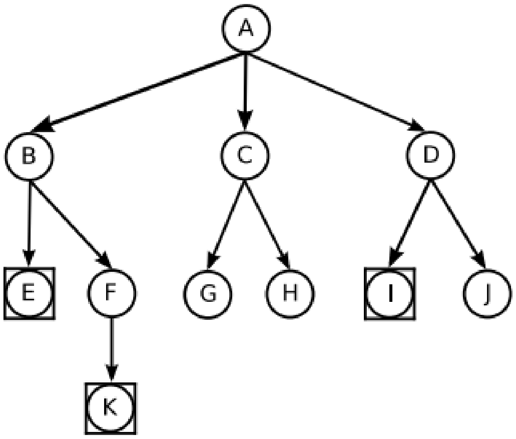


Figure 1: Search graph for task 1.

Fill out the following table:

Search Strategy	Order in which the nodes are taken from the frontier	Found goal
Breadth first search		
Depth First Search		
Iterative Deepening with increment 1		

SOLUTION:

1:

Here is the given table:

Search Strategy	Order in which the nodes are taken from the frontier	Found Goal
Breadth First Search (E)	A	-
	B, C, D	A
	C, D, E, F	A, B
	D, E, F, G, H	A, B, C
	E, F, G, H, I, J	A, B, C, D
	F, G, H, I, J	A, B, C, D, E
Breadth First Search (K)	A	-
	B, C, D	A
	C, D, E, F	A, B
	D, E, F, G, H	A, B, C
	E, F, G, H, I, J	A, B, C, D
	F, G, H, I, J	A, B, C, D, E
	G, H, I, J, K	A, B, C, D, E, F
	H, I, J, K	A, B, C, D, E, F, G
	I, J, K	A, B, C, D, E, F, G, H
	J, K,	A, B, C, D, E, F, G, H, I
	K	A, B, C, D, E, F, G, H, I, J
	-	A, B, C, D, E, F, G, H, I, J, K
Breadth First Search (I)	A	-
	B, C, D	A
	C, D, E, F	A, B
	D, E, F, G, H	A, B, C
	E, F, G, H, I, J	A, B, C, D
	F, G, H, I, J	A, B, C, D, E
	G, H, I, J, K	A, B, C, D, E, F
	H, I, J, K	A, B, C, D, E, F, G
	I, J, K	A, B, C, D, E, F, G, H
	J, K	A, B, C, D, E, F, G, H, I
Depth First Search (E)	A	-
	B, C, D	A
	E, F, C, D	A, B
	F, C, D	A, B, E
Depth First Search (K)	A	-
	B, C, D	A
	E, F, C, D	A, B
	F, C, D	A, B, E
	K, C, D	A, B, E, F
	C, D	A, B, E, F, K

Depth First Search (I)	A B, C, D E, F, C, D F, C, D K, C, D C, D G, H, D H, D D I, J J	- A A, B A, B, E A, B, E, F A, B, E, F, K A, B, E, F, K, C A, B, E, F, K, C, G A, B, E, F, K, C, G, H A, B, E, F, K, C, G, H, D A, B, E, F, K, C, G, H, D, I
Iterative Deepening with increment 1 (E)	1st iteration(d=0): T = 1: [A] 2nd iteration(d=1): T = 2: [A→B→C→D] 3rd iteration(d=2): T=3: [A→B→E]	E
Iterative Deepening with increment 1 (I)	1st iteration(d=0): T=1: [A] 2nd iteration(d=1): T=2: [A→B→C→D] 3rd iteration(d=2): T=3: [A→B→E→F→C→G→H→D→I]	I
Iterative Deepening with increment 1 (K)	1st iteration(d=0): T=1: [A] 2nd iteration(d=1): T=2: [A→B→C→D] 3rd iteration(d=2): T=3: [A→B→E→F→C→G→H→D→I→J] 4th iteration(d=3): T=4: [A→B→E→F→K]	K

2 Informed Search (31 Points):

Costs and a heuristic are now added to the search graph from the previous task. The corresponding graph is shown in Figure 2. The heuristic indicates the estimated distance to the closest target for a node and is shown in parentheses next to the node.

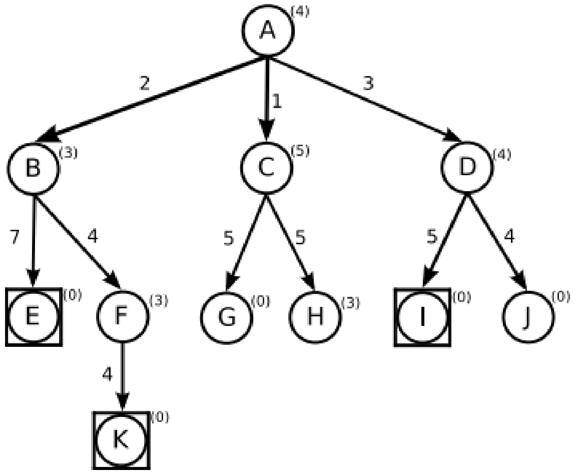


Figure 2: Search graph for task 2

1. In which order are the nodes taken from the frontier when searching using the A* algorithm. If two nodes with the same value are present in the frontier, the one which was added to the frontier first will be selected. Provide the f value for all nodes taken from the frontier. Which goal is found?

2. Suppose we modify the heuristic so that node D now has heuristic value (8). Which goal will be found by A*? Does A* still find the optimal solution?

SOLUTION:

2.1:

A: f(n)= 0+4= 4

A -> B : f(n) = 2+3 = 5

A -> C : f(n) = 1+5 = 6

A -> D : f(n) = 3+4 = 7

Among the above 3, A -> B corresponds to the least cost. Let’s explore A -> B

A -> B -> E : f(n) = 2+7+0 = 9

A -> B -> F : f(n) = 2+4+3 = 9

If we explore A -> C now, that gives

A -> C -> G: f(n) = 6

A -> C -> H: f(n) = 9

Both paths do not lead to any goal so we may omit these.

Among, A -> D , A -> B -> E and A -> B -> F, A->D gives the least cost so far so let’s explore this,

A -> D -> I : f(n) = 3+5+0 = 8

A -> D -> J : f(n) = 3+4+0 = 7

The only remaining path to be explored is now

A -> B -> F -> K : f(n) = 2+4+4+0 = 10

By looking at every path which led us to a target node, the path with the least cost is A -> D -> I with a f value of 8.

Hence Goal I is found.

2.2:

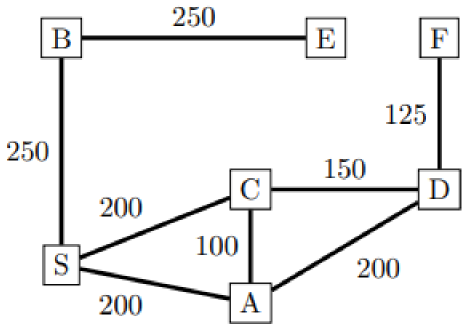
Altering D’s heuristic value to (8) does not change the goal. Because this heuristic value is not added when calculating the f(n) value for a target node. Only the corresponding weights of the edges and the n-th heuristic value is added, hence, heuristic values from preceding nodes do not affect the final cost whatsoever.

In the end, the solution remains the same even after the altercation.

3 Informed Search (45 Points):

Assume the following map with distances between nodes (cities). Your starting node is S. The target node is E. The following table contains estimations about the distance to E for each city:

Nodes	Estimated distance to E
S	400
A	350
B	240
C	150
D	160
E	0
F	50



Use these estimates as a heuristic for the following tasks:

a) In which order explores a Greedy-Best-First-Search (**without duplicate detection**) the graph, if the starting node is S and the target node is E. Explain your result.

b) Use an A*-Algorithm to find the shortest path between S and E. Use one row of the following table for each step. Always write down the node and its value according to f. Circle the node that is being taken out of the frontier in each step. Write down the path that the algorithm found. Also state whether the path is optimal. You can assume that each node is only expanded once, meaning that we assume a duplicate detection.

SOLUTION:

3.a:

Starting node is S and target node is E (given). Greedy-Best-First-Search explores nodes in the following order:

S, C, D, F, D, F, D, …….

We have run into an endless loop.

3.b:

Using an A* algorithm to find the shortest path between S and E and using one row of the given table for each step. Here also the node and its value are according to f. Here each node is only expanded once, meaning that we assume a duplicate detection:

(Removed elements are fat instead of circles)

Step	**Frontier/ Queue**			
1	**S = 400**			
2	A = 200+350 = 550	B = 250+240 = 490	**C = 200+150 = 350**	
3	A = 550	**B = 490**	D = 200+150+160 = 510	
4	A = 550	**E = 250+250+0 = 500**	D = 510	
5				
6				
7				
8				

The found path from S to E is optimal here (stated in the table).