Assignment 3 Comprehension Solutions

B351 / Q351

October 2, 2022

INSTRUCTIONS: Please submit your answers as ONE PDF file by the deadline in canvas. This is an individual assignment. Make sure you check your file to make sure you submitted what you intended to submit.

1 Comprehension Questions

1. Consider the following problem: You have two water containers measuring 5 and 3 gallons of water. Initially they are empty. You also have a water faucet. You can either fill up a container entirely, empty a container, or transfer water from one container to another. You must transfer all possible water. For example, if you had the 5 gallon container full of water, and wanted to transfer to an empty 3 gallon container, you must transfer three gallons. Your goal is to measure exactly 4 gallons of water (meaning in the end, one container should have 4 gallons of water). First, formulate a good representation of a state (ideally some sort of picture). You must then use your state representation to create a graph of the search space, starting with the initial state. Make sure to indicate the initial state as well as any goal states. **The should look similar to the kinds of problems we have been talking about in class.** Make sure this graph is legible. To help maintain legibility, feel free to draw a state more than one time to avoid a messy graph. If you do this, you **MUST** indicate that this state has been seen before. There is no need to further expand a state that has already been expanded once.

   Solution (25 pts.):
   State Space is defined/consistent (5 pts.):
   - - (neither bucket filled)
   n - (5-gallon bucket filled with n gallons)
   - m (3-gallon bucket filled with m gallons)
   n m (both buckets filled as above)
   Each bucket may only be filled fully from the source, or they may be filled from the other bucket (e.g., 3/5 - may be achieved by filling the 5-gallon bucket from the 3-gallon bucket)

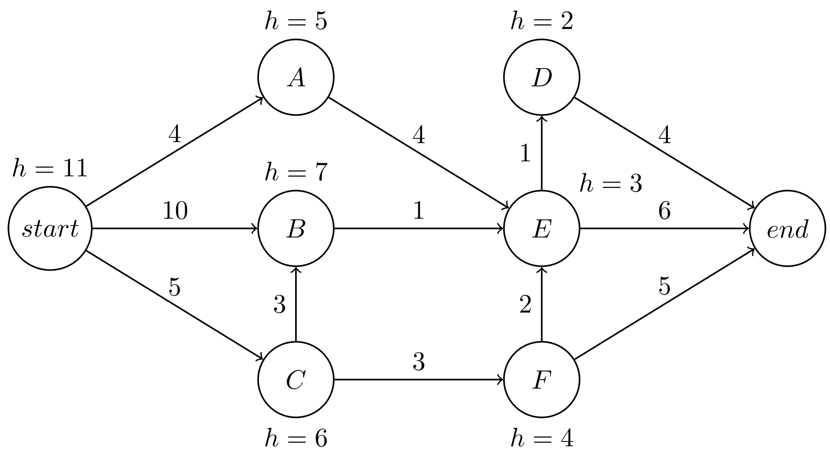
   State Space Graph (16 pts.):
   (-3 mostly correct, -6 missing major elements, -9 attempted) (state : child1 , child2 , etc.)
   - - : 5 - , - 3
   5 - : - - , 5 3 , 2 3
   - 3 : - - , 5 3 , 3 -

5 3 : 5 - , - 3
2 3 : 5 - , - 3 , 5 3 , 2 -
3 - : 5 - , - 3 , - - , 3 3
2 - : 2 3 , 5 - , - - , - 2
3 3 : 3 - , - 3 , 5 3 , 5 1
- 2 : - 3 , - - , 2 - , 5 2
5 1 : 5 3 , 5 - , - 1 , 3 3
5 2 : 5 - , 5 3 , - 2 , 4 3 (GOAL)
- 1 : - 3 , - - , 5 1 , 1 -
1 - : 1 3 , - - , 5 - , - 1
1 3 : - 3 , 5 3 , 1 - , 4 - (GOAL)

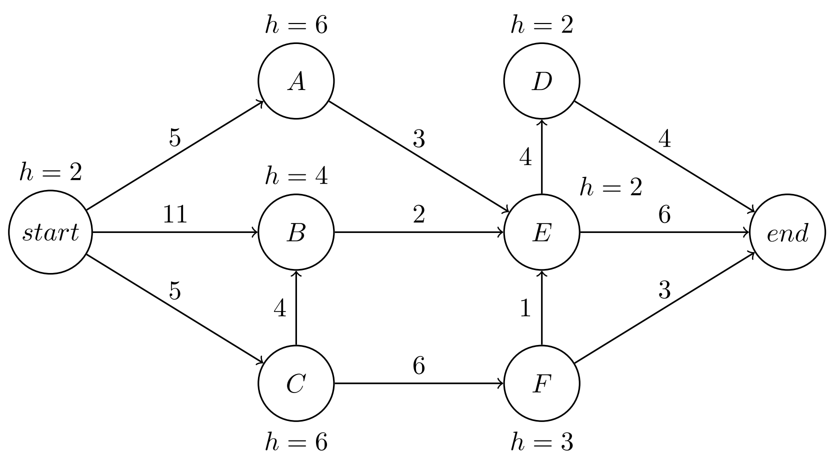
Possible solutions (4 pts. - 2 ea.):
- - , - 3 , 3 - , 3 3 , 5 1 , - 1 , 1 - , 1 3 , 4 -
- - , 5 - , 2 3 , 2 - , - 2 , 5 2 , 4 3

2. Determine whether the heuristics in the following graphs are admissible. Explain.

(a)



(b)



(15 pts) (a) 7.5 No the heuristic is not admissible because

$$h(B) = 7 \geq 6 = h^*(B)$$

(b) 7.5

$$h(start) = 11 \leq 13 = h^*(start)$$

$$h(A) = 6 \leq 9 = h^*(start)$$

$$h(B) = 4 \leq 8 = h^*(B)$$
$$h(C) = 6 \leq 9 = h^*(C)$$
$$h(D) = 2 \leq 4 = h^*(D)$$
$$h(E) = 2 \leq 6 = h^*(E)$$
$$h(F) = 3 \leq 3 = h^*(F)$$

The heuristic is admissible becuase $\forall_N, h(N) \leq h^*(N)$

3. Compare and contrast Dijkstra's algorithm `https://en.wikipedia.org/wiki/Dijkstra's_algorithm` to the A* search algorithm that we have discussed in lecture.

   (15 pts): (12 pts) 12 points for meaningfully discussing that dijkstra's calculates costs to all nodes, while A* calculates the cost to a target node, and that with a meaningful heuristic A* is faster at achieving its goal. Should mention that both can calculate the optimal cost to a target node (3 pts) 3 Remaining points for meaningfully including the definition of A* in some form, that is f(n) = g(n) + h(n). Essentially, they need to bring up the heuristic's effect on the results specifically for full points.

4. Consider the graph in Problem 2, part (b). The successors of a node are each node it points to. Step costs are given on each edge. Heuristic values are given next to each node (as $h = x$). For each search strategy below, show the order in which nodes are expanded(with the start node as the start state and the end node as the goal state) and the change of the fringe. Please end your paths with the goal node that is found, if any. Also show the path from start to goal and its cost, or write "None" if no path is found. Assume the successor function returns states in alphabetic order, e.g. the children of start are (A, B, C), which also means A is put on the fringe a ahead of B and B is put on the fringe ahead of C. Consider "end" to be at the start of the alphabet, and "start" to be at the end of the alphabet. (45 points)

   (a) DFS (7 points)

   Chosen Path: Start → C → F → E → D → End (1 point)
   Path Cost: 20 (1 point)
   Correct Expanded Nodes (2 points)
   Correct Fringe (3 points)

Node Expanded	**Fringe**	**Comments**
	{}	In DFS, the fringe is a last in first out queue.
	{Start}	
Start	{A,B,C}	Because C is put in last, so we expand C first.
C	{A,B,B,F}	Because F is put in last, so we expand F first.
F	{A,B,B,End,E}	Since "End" is considered start of Alphabet
E	{A,B,B,End,End,D}	E is removed before End
D	{A,B,B,End.End,End}	
End	{A,B,B,End,End}	Solution Found

Table 1

(b) BFS (7 points)

Node Expanded	**Fringe**	**Comments**
	[]	
	[Start]	
Start	[A,B,C]	
A	[B,C,E]	E Added
B	[C,E,E]	E Added
C	[E,E,B,F]	B,F Added
E	[E,B,F,End,D]	End,D Added (End is start of alphabet)
E	[B,F,End,D,End,D]	
B	[F,End,D,End,D,E]	
F	[End,D,End,D,E,End,E]	End,E Added
End	[D,End,D,E,End,E]	Solution Found

Chosen Path: Start $\rightarrow$ A $\rightarrow$ E $\rightarrow$ End (1 point)
Path Cost: 14 (1 point)
Correct Expanded Nodes (2 points)
Correct Fringe (3 points)

(c) Uniform Cost Search (8 points)

Node Expanded	**Fringe**	**New Entries**
	[]	a
		a
Start	A^5, C^5, B^{11}	A,C,B
A	C^5, E^8, B^{11}	E
C	E^8, B^9, B^{11}, F^{11}	B,F
E	$B^9, B^{11}, F^{11}, D^{12}, End^{14}$	D^{12}, End^{14}
B	$B^{11}, E^{11}, F^{11}, D^{12}, End^{14}$	E^{11}
B	$E^{11}, F^{11}, D^{12}, E^{13}, End^{14}$	E^{13}
E	$F^{11}, D^{12}, E^{13}, End^{14}, D^{15}, End^{17}$	D^{15}, End^{17}
F	$D^{12}, E^{12}, E^{13}, End^{14}, End^{14}, D^{15}, End^{17}$	E^{12}, End^{14}
D	$E^{12}, E^{13}, End^{14}, End^{14}, D^{15}, End^{16}, End^{17}$	End^{16}
E	$E^{13}, End^{14}, End^{14}, D^{15}, D^{16}, End^{16}, End^{17}$, End^{18}	D^{16}, End^{18}
E	$End^{14}, End^{14}, D^{15}, D^{16}, End^{16}, D^{17} End^{17}$, End^{18}, End^{19}	$D^{17}, End 19$
End		

Chosen Path: Start → A → E → End (1 point)
Path Cost: 14 (1 point)
Correct Expanded Nodes (3 points)
Correct Fringe (3 points)

(d) (Greedy) Best-First Search with $f(N) = h(N)$ (7 points)

Node Expanded	**Fringe**	**New Entries**
	[]	a
Start	B^4, A^6, C^6	A,B,C
B	E^2, A^6, C^6	E^2
E	$End^0, D^2, E^2, A^6, C^6$	D^2, End^0
End		

Chosen Path: Start → B → E → End (1 point)
Path Cost: 19 (1 point)
Correct Expanded Nodes (2 points)
Correct Fringe (3 points)

(e) Iterative Deepening Search (8 points)

Chosen Path: Start → C → F → End 1 (point)
Path Cost: 14 (1 point)
Depth 1: Expanded Nodes + Fringe (2 points)
Depth 2: Expanded Nodes + Fringe (2 points)
Depth 3: Expanded Nodes + Fringe (2 points)

Node Expanded	**Fringe**	**Comments**
	{}	In DFS, the fringe is a last in first out queue.
	{Start}	
Start	{A,B,C}	Because C is put in last, so we expand C first.
C	{A,B,}	B and F won't added as it will violate depth constraint
B	{A}	
A	{}	Solution Not found

Table 2: IDS: Depth 1

Node Expanded	**Fringe**	**Comments**
	{}	In DFS, the fringe is a last in first out queue.
	{Start}	
Start	{A,B,C}	Because C is put in last, so we expand C first.
C	{A,B,B,F}	
F	{A,B,B}	Depth Limit Reached
B	{A,B}	Depth Limit Reached
B	{A,E}	This time limit not reached.
E	{A}	
A	{E}	
E	{}	No solution found

Table 3: IDS: Depth 2

Node Expanded	**Fringe**	**Comments**
	{}	In DFS, the fringe is a last in first out queue.
	{Start}	
Start	{A,B,C}	Because C is put in last, so we expand C first.
C	{A,B,B,F}	
F	{A,B,B,End,E}	Since, End is start of alphabet
E	{A,B,B,End}	Depth Limit Reached
End	{A,B,B}	Solution Found

Table 4: IDS: Depth 3

(f) A* Search (8 points)

Chosen Path: Start → A → E → End 1 (point)
Path Cost: 14 (1 point)
Showing all three depths (1 point)
Correct Expanded Nodes (3 points)
Correct Fringe (3 points)

Node Expanded	**Fringe**	**New Entries**
	[]	
Start	A^{11}, C^{11}, B^{15}	A,B,C
A	E^{10}, C^{11}, B^{15}	
E	$C^{11}, End^{14}, D^{14}, B^{15}$	
C	$B^{13}, End^{14}, D^{14}, F^{14}, B^{15}$	
B	$E^{13}, End^{14}, D^{14}, F^{14}, B^{15}$	
E	$End^{14}, D^{14}, D^{14}, B^{15}, End^{17}, D^{17}$	Note 1
End		

Note 1: Both End and D are 17, but end gets preference as it is start of alphabet

2 Bonus Problem (10%)

1. Please describe in your own words why it is important to use an admissible heuristic with the A* search algorithm. Additionally, create a graph and two separate heuristic functions. One heuristic should be admissible and the other should not be admissible. Use the A* search algorithm to expand each path like you did in problem 3 using each heuristic. **Design your example such that running the algorithm with the admissible heuristic results in the optimal solution, and running it with the other heuristic does not.**

Solution:
Important because admissible means that f < f* and therefore f(optimal path to goal) < f*(optimal path to goal) < f*(suboptimal goal) = f(suboptimal goal). (4pts)
h < h* or f < f* - 1pts
won't expand suboptimal goal node - 1pts
will find optimal solution if it exists - 1pts
any attempt at details of proof - 1pts
exercise (6pts)
single graph including start, goal, and at least one other node. 1pts
heuristic A admissible 1pts

heuristic B inadmissible 1pts
proof of admissibility 1pts and inadmissibility 1pts
A* search 1pts