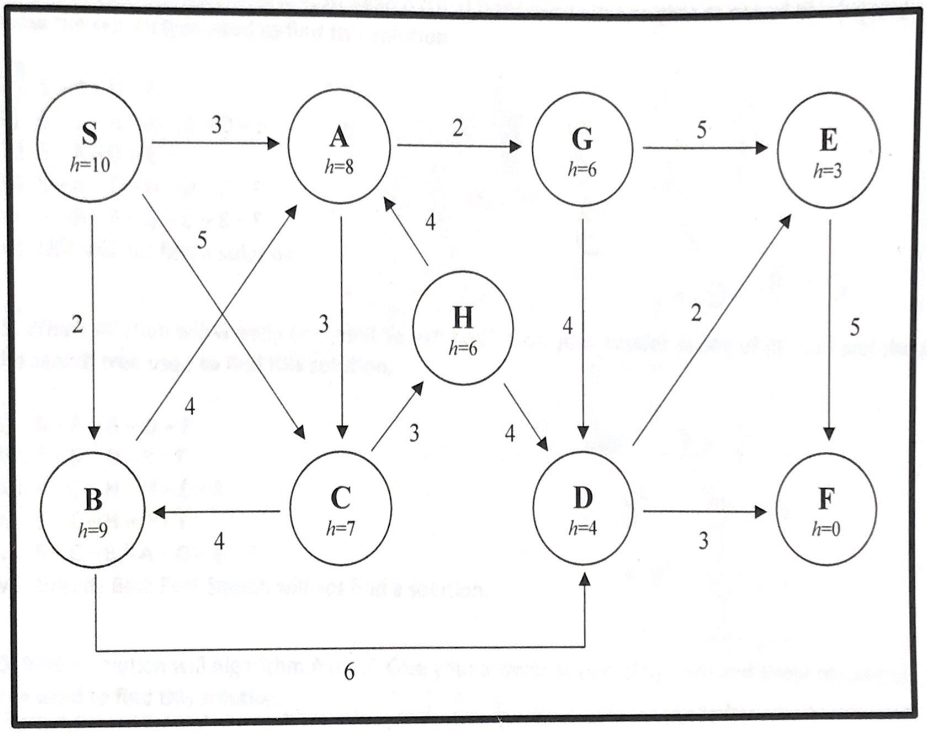
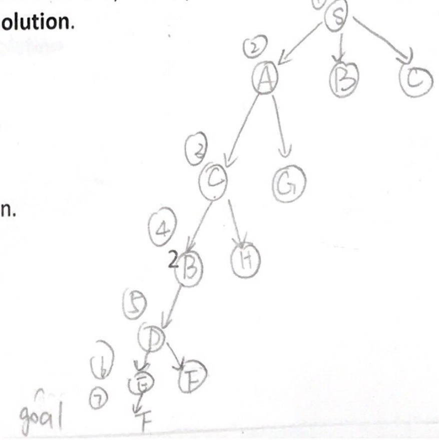
Problem 1: Search Algorithms [25 points]

You are given below a state-space graph that consists of **nine** states, the **costs** of the connections between them, and a **heuristic**, ***h*(*n*)**, for each state. Your task is to find a path from start state **S** to goal state **F**. In order to find a solution path, one can use a number of different search methods. In the following questions, you are to find the path from **S** to **F** that the search algorithm given in the question would yield. Use the **tree-search algorithm** given in Figure 3.7 in the textbook where the **goal test** is performed when a state is removed from ***Frontier***. Assume that states are selected/expanded in **alphabetical** order when a **tie** occurs (e.g., if there is a tie between states A and B, then expand A first).. **Repeated states** along a path from a node back to the root are **not** allowed. Lastly, if there happen to be several instances of the *same* state in ***Frontier*** when expanding (i.e., two of the same states that have different paths back to **S**), expand first the one that has been in ***Frontier*** longest.



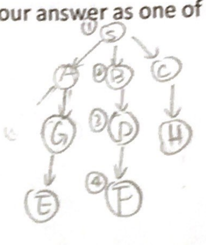
(a) [5] Which solution path will the Depth-First Search (DFS) algorithm find? Expand the successors of a node in alphabetical order (e.g., if a node has 3 successors, A, B, and C, then A will be expanded before B, and B will be expanded before C). Give your answer as one of (i) – (vi) **and show the search tree used to find this solution.**

[✓ (i) S – A – C – B – D – E - F]
(ii) S – A – C – H – D – F
~~(iii) S – A – C – B – D – F~~
(iv) S – B – A – G – D – E – F
(v) S – C – B – A – G – D – F
(vi) DFS will *not* find a solution.



(b) [5] Which solution path will the Breadth-First Search (BFS) algorithm find? Expand the successors of a node in alphabetical order (e.g., if a node has 3 successors, A, B, and C, then A will be expanded before B, and B will be expanded before C). Give your answer as one of (i) – (vi) **and show the search tree used to find this solution.**

(i) S – A – G – E – F
(ii) S – C – H – D - F
(iii) S – B – D – F
(iv) S – A – G – D – F
(v) S – A – G – D – E – F
(vi) BFS will *not* find a solution.

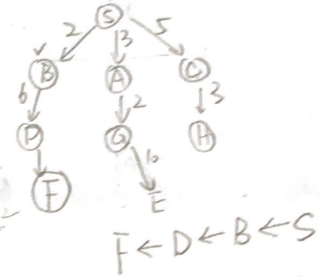


(c) [5] Which solution will Uniform-Cost Search (UCS) find? Give your answer as one of (i) – (vi) **and show the search tree used to find this solution.**

[✓ (i)] S – B – D – F
(ii) S – C – H – A – G – D – F
(iii) S – B – D – E – F
(iv) S – A – C – H – D – E – F
(v) S – B – A – G – D – E – F
(vi) UCS will *not* find a solution.

g(n)

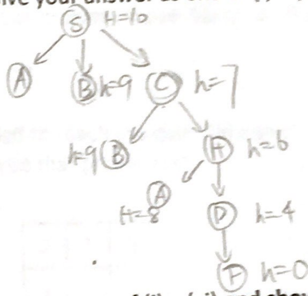
S	B2 A3 C5
B	A3 C5 D8
A	C5 G5 D8
C	G5 D8 H8
G	D8 H8 E10
D	F11, F15,
F	F11 F15



(d) [5] Which solution will Greedy Best-First Search find? Give your answer as one of (i) – (vi) **and show the search tree used to find this solution.**

f=h(n)

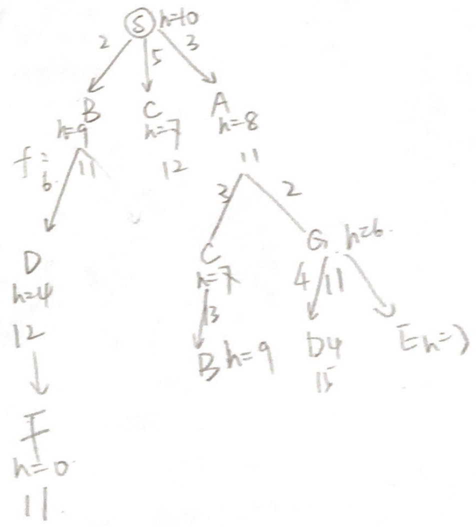
(i) S – A – G – D – F
(ii) S – B – D – E – F
(iii) S – C – H – D – E – F
(iv) S – C – H – D – F
(v) S – C – B – A – G – E – F
(vi) Greedy Best-First Search will *not* find a solution.



(e) [5] Which solution will Algorithm A find? Give your answer as one of (i) – (vi) **and show the search tree used to find this solution.**

F=g+h

[✓ (i)] S – B – D – F
(ii) S – A – C – B – D – E – F
(iii) S – A – G – D – F
(iv) S – A – C – B – D – F
(v) S – B – D – E – F
(vi) Algorithm A will *not* find a solution.


S h=10
2
5
3
B h=9
C h=7
A h=8
f = 6
11
12
11
3
2
D h=4
12
C h=7
G h=6
4
11
3
F h=0
11
B h=9
D4
15
E h=)