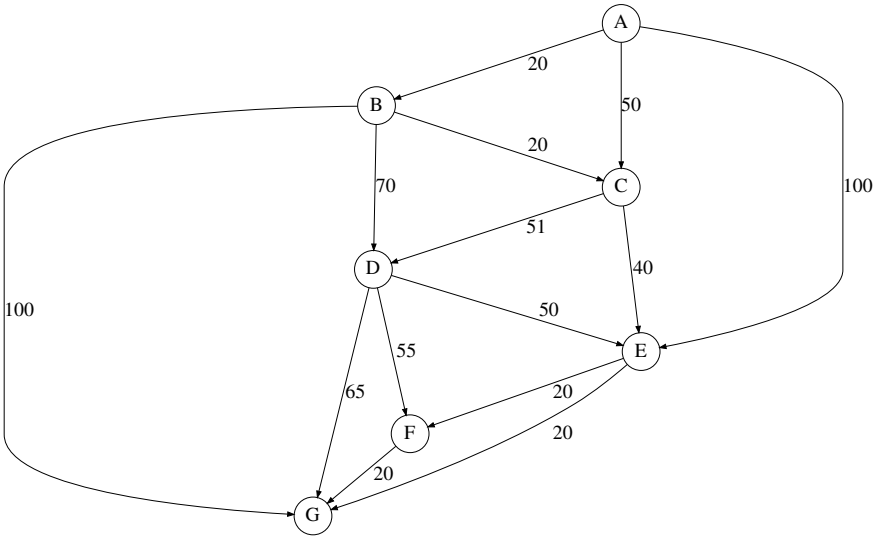
CSCE 625-700 Homework #2

Upload scanned handwritten solution or typed pdf file to Canvas.

Informed Search

Total: 100 points



Node	h(n)
A	20
B	28
C	32
D	53
E	14
F	18
G	0

Figure 1: **Informed Search.**

Problem 1 (Written; 10 pts): For the problem shown in Fig. 1, show that the heuristic is admissible ($h(n) \leq h^*(n)$ for all n). Note: You have to compute $h^*(n)$ for each n and compare to the $h(n)$ table.

Hint: It is best to work backwards from the goal, where $h^*(G) = 0$ (already at goal), $h^*(F) = 18$ (true minimum cost from (F) to (G)), $h^*(E) = 20$ (true minimum cost from (E) to (G)), etc.

Problem 2 (Written; 20 pts): Manually conduct greedy best-first search on the graph below (Fig. 1), with initial node (A) and goal node (G). Actual cost from node to node are shown as edge labels. The heuristic function value for each node is shown in a spearate table to the right. Show:

1. Node list content at each step
2. Node visit order
3. Solution path

4. Cost of the final solution.

Problem 3 (Written; 20 pts): (1) Repeat the problem right above with A* search. (2) In addition, show the $f(n)$ value for all nodes expanded (you need this to sort them in the node list). (3)Which one gives a lower cost solution: Greedy best-first or A*?

Note: Note that the same node can appear in the node list with a different $f(n)$ value, depending on the path taken. For example, $f(C)$ will be different if you followed a different path to reach the node: $A \rightarrow B \rightarrow C$ (where $f(C) = 20 + 20 + 32 = 72$) vs. $A \rightarrow C$ (where $f(C) = 50 + 32 = 82$). Due to this, you may need to track which path you followed to reach node n and calculate the $f(n)$ value accordingly. It helps to write the node C as ${}_{AB}\textcircled{C}$ to indicate the path in the subscript ($A \rightarrow B \rightarrow C$). Also, for sorting, it helps to indicate the f value as a subscript. For example, ${}_{AB}\textcircled{C}_{72}$.

Note: When sorting the node list by hand, some tied values may appear. In this case, put the oldest node ahead of new nodes.

Problem 4 (Programming; 25 pts): Using the `dfs.ipynb` code, implement greedy-best-first search, and solve the problem in (Fig. 1). Print out the same output as required by Problem 2.

You are required to use the provided starter code and the data structure. You are not allowed to inven t your own data structure and function interface.

Use https://colab.research.google.com.

Problem 5 (Programming; 25 pts): Using the `dfs.ipynb` code, implement A* search, and solve the problem in (Fig. 1). Print out the same output as required by Problem 3.

Use https://colab.research.google.com.

IMPORTANT: For Problem 4 and 5, to get full credit, you must submit the screenshot of your pinned revision (colab) or revision history. See Canvas instructions for details.