Name: ______________________ *ID:* _________________________

"If You only do what you can, you will never be more than you are now." – Master Shifu

Search Algorithms

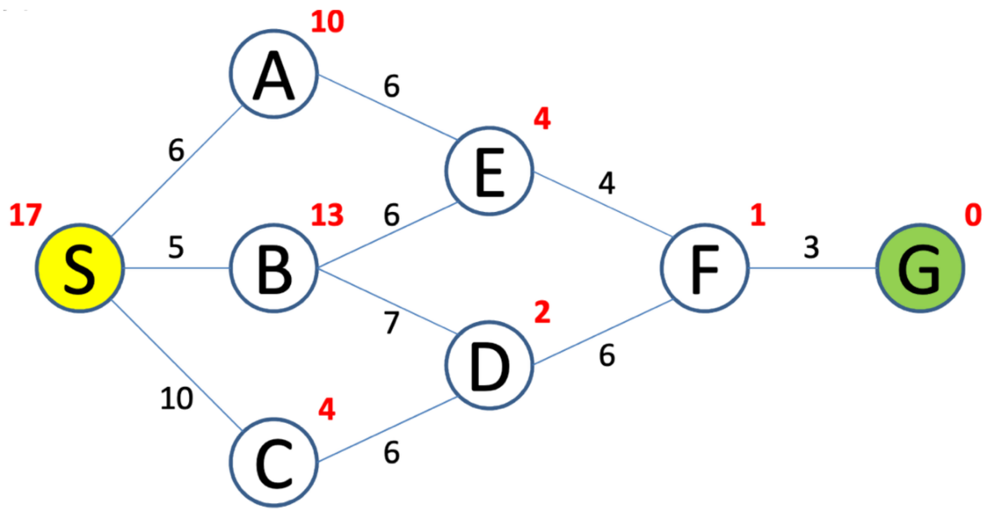
Instructions:

- *This written task is being released as a subpart of Assignment 1 in which you'll get hands on experience in Search Algorithms in Python. The written task will provide the foundations for developing strong theoretical background in 'Search Algorithms' which is a prerequisite for programming part of this assignment. The programming part will be released soon; You should use this opportunity (part 1) to attain as much theoretical knowledge on Search algorithms as possible. We may release part 2 any time soon in parallel to part 1, and you should try to be done with part 1 at your earliest. Yet, we have generously set up the* ***deadline*** *for part 1 of* ***February 16, 2020****.*
- *The questions in this task do not carry equal weightage.*
    - *For most of the questions, you may find section 3.4 of your textbook helpful.*
    - *For questions on A* Search, you may find section 3.5 of your textbook helpful.*
        - *Textbook: Artificial Intelligence: A Modern Approach (3rd Edition).*
        - **"It is what you read [*Artificial Intelligence: A Modern Approach]* when you don't have to that determines what you will be when you can't help it."**- Oscar Wilde
- *We are looking for precise answers.*
- *We are looking for 'queues' and 'sets' in tables at each step in Q-9, 10, 11 and 14.*
- *Write your answers in Word; Submit a hardcopy and upload the word file and the pdf file on LMS.*
- *Respect and abide by the honor code of LUMS university.*

Questions

1. Briefly compare Breadth-First Search (BFS) and Depth First Search (DFS).
2. What issue does Depth limited Search resolve?
3. Precisely describe A* search Algorithm.
4. How does uniform cost search differ from A* search?
5. Which of the four algorithms mentioned so far is(are) complete? Which is(are) optimal?
6. Would you use a 'Last in, first out' LIFO queue (also called a stack) or a 'First in, first out' FIFO queue for implementing DFS in your assignment 1? What would your choice be for BFS?
7. Describe the most important property for the heuristic used in A* search.

The following questions are based on the figure shown:



S denotes the start node and G denotes the goal node.

8. What is the optimal path in the graph?
9. Perform BFS on the graph shown. Explicitly show queue at each step along with a set of explored nodes so far. Are you using a LIFO queue? Which path is found by BFS, is it optimal? Write the expanded nodes in sequence. You may ignore the costs shown for this question.
10. Now, Perform DFS on the graph shown. Explicitly show queue at each step along with a set of explored nodes so far. Are you using a LIFO queue? Which path is found by DFS, is it optimal? Write the expanded nodes in sequence. You may ignore the costs shown for this question.
11. Perform A* Search on the graph shown. Explicitly show queue at each step along with a set of explored nodes so far. Write the expanded nodes in sequence.

To get you started- for questions 9, 10 and 11, we have provided you the queue at step 0 when the start node S is expanding- as well as a set of explored nodes. As already stated, you may remove the second row of costs f(n) for questions: 9 and 10; You'll need it for A search. Be careful, when you think it's time to stop your algorithm. You may find pseudocodes in your textbook helpful- specifically in figure 3.11 for Breadth-first Search, figure 3.14 for Uniform Cost Search, and figure 3.7 for the general graph search.*

Expanding Node from *Queue*: *S*

Nodes in Queue	A	B	C		
f(n) = g(n) + h(n)	6 + 10 = 16	5 + 13 = 18	10 + 4 = **14**		

Set of Explore Nodes

Nodes Explored	S							

12. Did A* search find the optimal path when you performed it on the graph?
13. If A* found the optimal path, what two properties the graph has which guaranteed the optimality of A*. On the other hand, if the A* failed to find the optimal path, which property/properties the graph lack(s)?
14. If A* failed to provide the optimal path in the graph above, change the heuristic values such that A* guarantees the optimal path. To validate, perform A* search again (as you did in Q-11) with new heuristics costs. You may change only the heuristic costs. Costs to reach a node *n* from start node *S* should remain unchanged. Note that you are free to choose any heuristic costs as long as A* guarantees optimality.
15. Precisely describe genetic algorithms.

Good Luck!

Spoiler Alert!