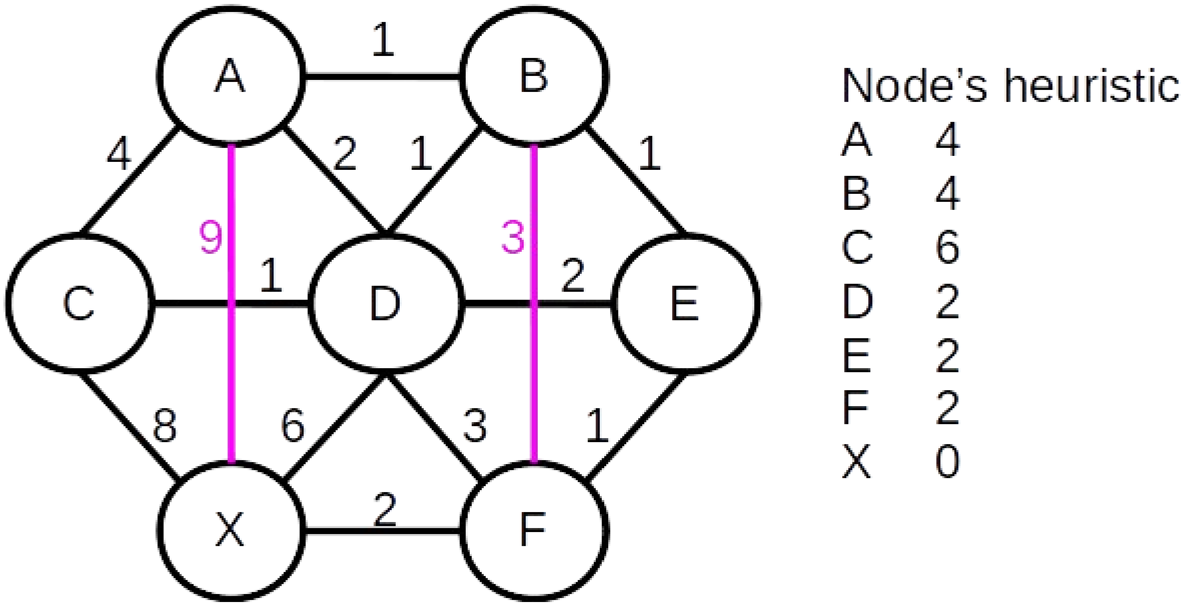
4511W, Fall-2021
ASSIGNMENT 2:
Assigned: 09/28/21 Due: 10/05/21 at 11:55 PM (submit on gradescope, mark the page(s) associated for each problem when submitting)

Problem 1. (20 points)
Run A* search on the following graph with initial state “A” and goal state “X”. At every step indicate the fringe and explored lists (with both the node and f-cost). (Note: pink color just to help associate edges & weights/costs.)



Problem 2. (36 points)
Suppose you had a tree with a branching factor of 3. The maximum depth of the tree is 5. Find the best-case and worst-case number of nodes searched (for various goal depths) for the following three algorithms: breadth-first-search, depth-first-search, and iterative-deepening depth-first-search. Show sufficient work that we can understand the reasoning for your answer.

Note: this is **not** the big-O notation, but the specific count for best/worst case.

2.1. If the goal is at depth 3, what is minimum amount of nodes searched for BFS?
2.2. If the goal is at depth 3, what is maximum amount of nodes searched for BFS?
2.3. If the goal is at depth 3, what is minimum amount of nodes searched for DFS?
2.4. If the goal is at depth 3, what is maximum amount of nodes searched for DFS?
2.5. If the goal is at depth 3, what is minimum amount of nodes searched for ID-DFS?
2.6. If the goal is at depth 3, what is maximum amount of nodes searched for ID-DFS?

2.7. If the goal is at depth 4, what is minimum amount of nodes searched for BFS?
2.8. If the goal is at depth 4, what is maximum amount of nodes searched for BFS?
2.9. If the goal is at depth 4, what is minimum amount of nodes searched for DFS?
2.10. If the goal is at depth 4, what is maximum amount of nodes searched for DFS?

2.11. If the goal is at depth 4, what is minimum amount of nodes searched for ID-DFS?
2.12. If the goal is at depth 4, what is maximum amount of nodes searched for ID-DFS?

Problem 3. (14 points)
In class I stated (also in the book) that “if a heuristic is consistent, then it **must** be admissible” also “if a heuristic is **not** admissible, then it **cannot** be consistent”. Prove (formally) this second part that “not admissible means not consistent” (assume h(goal)=0). There can be non-numerical parts of the proof, but the logic needs to be sound.

Problem 4. (10 points)
Suppose you have a graph that you want to search.
4.1. What is one issue with running DFS as a graph-search on this problem?
4.2. What is one issue with running DFS as a tree-search on this problem?

AIMA code:
Problem 5. (20 points)
5.1. Test bi-directional, breadth first and A* search (in the AIMA code) on a number of different 8-puzzles (try enough variations to be confident in your answer) and generalize which would work best (in terms of runtime).

Like last time, I suggest that you use /tests/test_search.py as a reference, despite being unable to run this. (The bi-directional search is a bit less polished than the other code, but you should be able to run it, though getting things like the solution are difficult, though not necessary for this problem.)

5.2. Also answer: what sub-search is being done from both initial and goal state in the implemented bi-directional search in the given AIMA code (you will need to poke around a bit in search.py)?

(Note: some arrangement of numbers in an 8-puzzle are impossible to solve.)