CS 411: Artificial Intelligence I
Spring 2018
Homework 1
Due: February 11th, 11:59pm (via Blackboard)

Programming Portion

This portion of the assignment may be completed individually or in groups of 2.

Complete Project 1 at: `http://ai.berkeley.edu/search.html`, Questions 1–8.

Your code for files `search.py` and `searchAgents.py` should be submitted to Blackboard for evaluation. **It must be your own (or you and your partner's) code and should not be copied from any other source.** We will check for similarity to other submissions and existing resources available on the web for any cheating.

Hints: (i) Graph search is almost always better than tree search. (ii) Implement your closed list as a dict or set! (iii) Nodes are conceptually paths, but better to represent with a state, cost, last action, and reference to the parent node

Written Portion

This portion of the assignment must be completed individually.

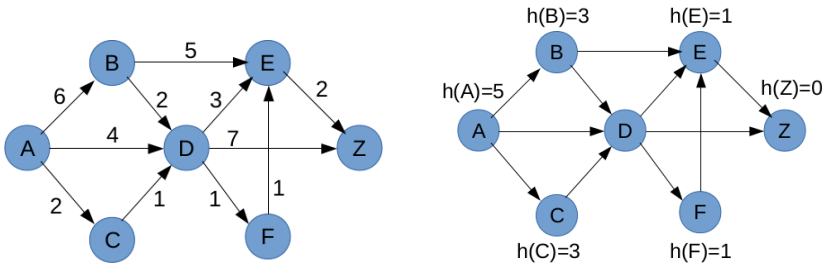

Figure 1: Search cost graph (left) and heuristic function for reaching goal Z (right).

1. **Search tree (3 points).** Draw the complete search tree for this graph starting from A. Please list children in the tree in alphabetical order from left to right.
2. **Uniform cost tree search (2 points).** Indicate which nodes of the search tree will be explored using uniform cost tree search from A to Z.
3. **Uniform cost graph search (2 points).** Indicate which nodes of the search tree will be explored using uniform cost graph search from A to Z.
4. **Heuristic Admissibility (1 point).** The heuristic value for D, $h(D)$ has not been provided. What range of values can $h(D)$ have and still be admissible?
5. **Heuristic Consistency (2 points).** What range of values can $h(D)$ have and still be consistent?
6. **A* tree search (3 points).** Indicate which nodes of the search tree will be explored using A* tree search and the provided heuristic function with $h(D) = 2$.
7. **A* graph search (3 points).** Indicate which nodes of the search tree will be explored using A* graph search and the provided heuristic function with $h(D) = 2$.
8. **Reverse search tree (3 points).** Sometimes searching backward can be more efficient. Reverse the direction of the edges of the search cost graph and show the resulting search tree from Z to A.
9. **Reverse uniform cost tree search (2 points).** Indicate which nodes of the reverse search tree will be explored using uniform cost tree search from Z to A.
10. **Bidirectional uniform cost tree search (4 points).** Another useful trick is to search forward (A to Z) and backward (Z to A) simultaneously until a node is explored from both directions. Show the portions of the forward and backward search trees that are explored to find this common node.