University of Waterloo
Department of Electrical and Computer Engineering
ECE 457A Adaptive and Cooperative Algorithms

Due Date May 30, 2016 11:59 PM

Question 1:

(**State space formulation**) You are given two jugs – a 4 gallon jug and a 3 gallon jug. Neither jug has any measuring marks on it. There is a pump that can be used to completely fill one or both jugs. You can pour water from one jug into the other or onto the ground. The problem is to get exactly 2 gallons of water into the 3 gallon jug.

Determine a suitable state space formulation for the problem. For the chosen representation, how is initial state "both jugs are empty" represented? How is the goal state represented? List the set of actions (along with any conditions necessary for the action to be applied) that can be performed in order to solve the problem.

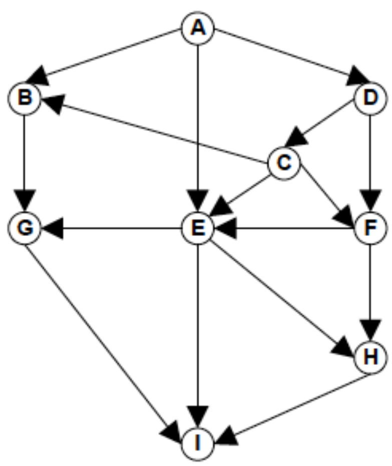
Question 2:

(**State space formulation**) Assume you have 3 poles and 64 discs of decreasing diameter. A disc may be stacked only on top of a larger disc or onto an empty pole. Initially, all discs are stacked correctly on one of the poles. The objective is to correctly stack all of the discs on a different pole.

Determine a suitable state space representation for the problem. How are the initial state and goal states represented? Define a set of actions (along with any conditions necessary for the action to be applied) that can be performed in order to solve the problem.

Question 3:

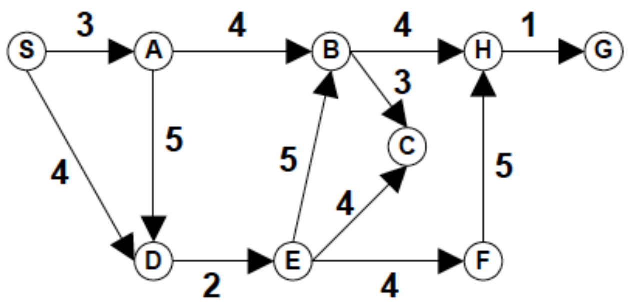
(Uninformed search) Consider the state space graph below in which the initial state is "A" and the goal state is "I". All actions that cause movement from one state to another state are shown with arrows and have a cost of 1. Assume the following: (i) when a state is expanded, its children are generated in alphabetical order, (ii) repeated states are ignored and (iii) goal testing is not performed when generating children.



Apply both Breadth-First Search and Depth-First Search to find a path from "A" to "I". Illustrate the contents of the open and closed queues for both algorithms at each step of the search.

Question 4:

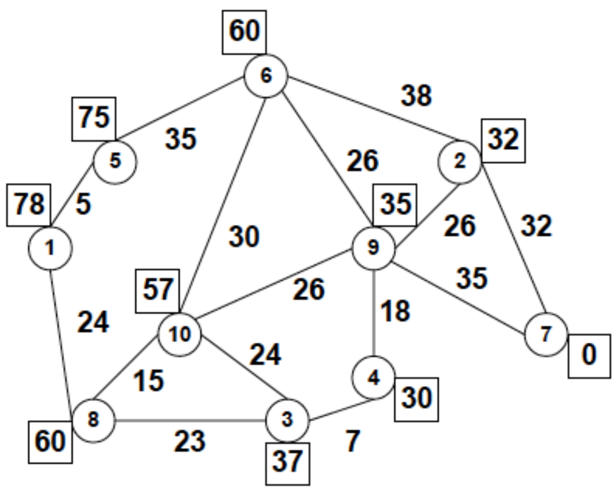
(Uninformed search) Consider the following state diagram with initial state "S" and goal state "G". Edges represent actions and the labels on the edges indicate the cost of the actions.



Assuming alphabetical order is used to break ties during the generation of children, list the sequence of nodes visited (i.e., illustrate the contents of the open and closed queues) and the final paths returned using Breadth-First Search, Depth-First Search and Uniform Cost Search.

Question 5:

(Informed search) Consider the city map below. The objective is to determine the shortest distance from city 1 to city 7.



Edges (which represent valid paths of travel between cities) are shown between cities and are labeled with the actual travel distance. Numbers shown in boxes beside each city represent an estimate of the distance from that city to city 7. Apply (i) Uniform Cost Search, (ii) Greedy Best First Search and (iii) A* Search to determine a path from city 1 to city 7. Show the contents of the open and closed queues during each step of each algorithm. Break any ties according the numerical value of the city (smaller cities take precedence).

Question 6:

(**Heuristic functions**) Consider the problem of finding paths in a 2-dimensional maze. Let g(n) be the path cost and h1(n) and h2(n) be two different heuristic functions. Specifically, let h1(n) be the Euclidean distance from the some position n to the goal position and let h2(n) be the Manhattan distance from some position n to the goal position. Both h1(n) and h2(n) are admissible functions. Given this information, answer the following questions:

a) Is h1(n) more informed that h2(n)?
b) Is h(n) = max{h1(n),h2(n)} admissible?
c) Is h(n) = min{h1(n),h2(n)} admissible?
d) Is h(n) = h1(n) + h2(n) admissible?
e) Is h(n) = h2(n) – h1(n) admissible?
f) Let h(n) = 2.5*h2(n) and prove that h(n) is not admissible.

Question 7:

(**Heuristic functions**) Consider the sliding tile puzzle which consists of 3 black tiles (B), 3 white tiles (W) and one blank tile (_). The initial configuration for the puzzle is shown below.

B	B	B	_	W	W	W

A tile can be moved into an adjacent blank position with a cost of 1. A tile can jump over one or two other tiles into a blank position with a cost equal to the number of tiles jumped over. The goal is to have all the white tiles to the left or all the black tiles. The position of the blank tile is not important.

Propose a heuristic function for solving this problem and analyze it with respect to its admissibility, monotonicity and informedness.