CS 461 – Artificial Intelligence (SPRING 2021)

Dr. Rabia Maqsood

Assignment # 1

Topics Covered: Problem formulation, task environment and PEAS, State space, Uninformed search algorithms **Group Assignment: only 2 persons**	Submission Deadline: Monday, March 29, 2021 by 16.00 sharp *Submit hand-written solution for P#1-3 and your source file for P#4 on SLATE.*

Problem # 1: The **missionaries and cannibals** problem is usually stated as follows. Three missionaries and three cannibals, along with one boat that can hold at most two people, are on the left bank of a river. The most salient thing about missionaries and cannibals in "cohabitation" is that if ever the cannibals in any one spot (left bank, right bank, on the boat) outnumber the missionaries, the outnumbered missionaries will be consumed – eaten! The goal of this problem is to get all six individuals safely across the river from the left bank to the right bank.

(a) **Formulate the problem** precisely (as discussed in the class), making only those distinctions necessary to ensure a valid solution.

(b) Draw a diagram of the complete **state space**.

Problem # 2: For each of the following activities, (a) give a **PEAS** description of the task environment (in a tabular form), and (b) characterize each of them in terms of the **task environment properties** listed below (create another table) and briefly explain your answer.

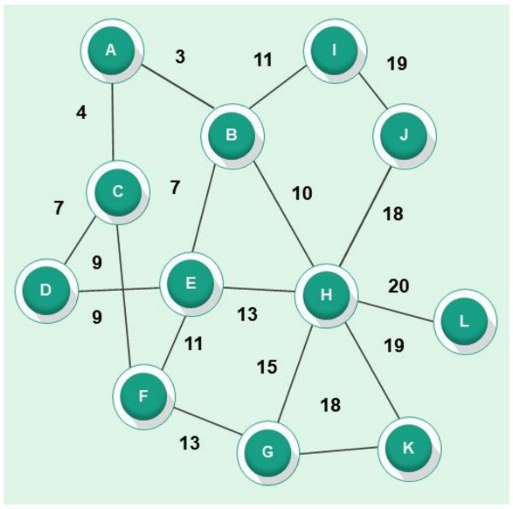
Fully observable or partially observable	Deterministic, stochastic or strategic
Sequential or episodic	Static, dynamic or semi-dynamic
Discrete or continuous	Single or multi-agent

i. Playing soccer.
ii. Exploring the subsurface oceans of Titan.
iii. Shopping for used AI books on the Internet.
iv. Playing a tennis match.
v. Practicing tennis against a wall.
vi. Performing a high jump.
vii. Knitting a sweater.
viii. Bidding on an item at an auction.

Problem # 3: Give a complete **problem formulation** for each of the following problems statements. Choose a formulation that is precise enough to be implemented.

(a) There are six glass boxes in a row, each with a lock. Each of the first five boxes holds a key unlocking the next box in line, the last holds a banana. You have the key to the first box, and you want the banana.
(b) You start with the sequence ABABAECCEC, or in general any sequence made from A, B, C, and E. You can transform this sequence using the following equalities: AC = E, AB = BC, BB = E, and Ex = x for any x. For example, ABBC can be transformed into AEC, and then AC, and then E. Your goal is to produce the sequence E.
(c) There is an n by n grid of squares, each square initially being either unpainted floor or a bottomless pit. You start standing on an unpainted floor square, and can either paint the square under you, or move onto an adjacent unpainted floor square. You want the whole floor painted.
(d) d. A container ship is in port, loaded high with containers. There 13 rows of containers, each 13 containers wide and 5 containers tall. You control a crane that can move to any location above the ship, pick up the container under it, and move it onto the dock. You want the ship unloaded.

Problem # 4: In this problem, you need to write a program to implement the following **uninformed search algorithms**: Breadth-first search, Depth-first search, Depth limited search and Uniform-cost search. For simplicity, we pick the following weighted graph to be implemented as an example.



Your program should provide the following features.

(a) Input start (S) state, goal (G) state and the algorithm to be executed from the user.
(b) Find and display a path between S and G (if possible) using the selected algorithm.
(c) Once the search algorithm stops, your program should display the followings.
    a. Complete path with states visited from S to G.
    b. Total no. of nodes placed on the frontier (i.e., Queue, Stack, etc.).
    c. Total no. of nodes generated during the path search.
    d. Total path cost (assume the step cost is 1 for algorithms which do not consider the weight between states for selection).

Tie breaking: For all algorithms, if there are multiple equivalent choices about which node goes next, your program should pick the node that comes first in **alphabetic** order (i.e. chronological order).

Implementation details: You may choose any programming language for implementation, e.g., Python, Java or C++. And, if you wish to use some other language, discuss with your course instructor first. Furthermore, you are expected to write all the code by yourself including the graph structure and other data structures that you might need to use for running an algorithm.