National University of Singapore
School of Computing
CS3243 Introduction to AI

Tutorial 7: Mid Term Review

Issued: March 11, 2021 Discussion in: Week 9

Important Instructions:

- **Assignment 7** consists of **Question 10** from this tutorial.
- Your solutions for the above questions must be TYPE-WRITTEN.
- You are to submit your solutions on LumiNUS by **Week 8, Saturday, 2359 hours**.
- Refer to LumiNUS for submission guidelines

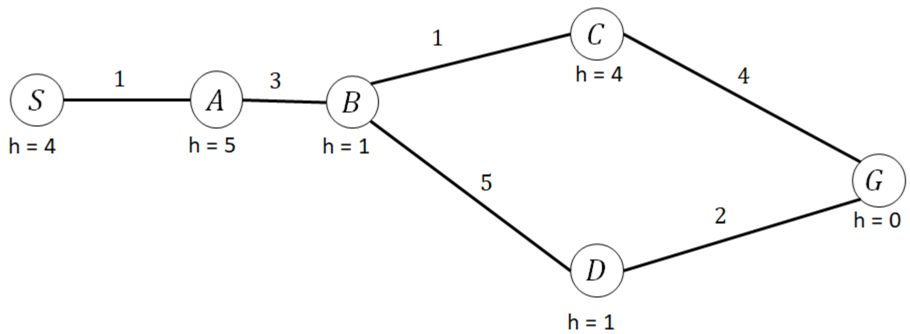
Note: you may discuss the content of the questions with your classmates, but you must work out and write up your solution individually - solutions that are plagiarised will be penalised.

---

1. Which of the following algorithms returns the optimal path if all step costs are constant? Select all that are applicable.
   (a) Breadth-First Search
   (b) Uniform-Cost-Search
   (c) Depth-First Search
   (d) Depth-Limited Search
   (e) Iterative Deepening Search
2. Consider the graph below where node S is the start node and node G is the goal node. There exist a node where its heuristic value is incorrect.

   Determine the node in which its heuristic value is incorrect such that we can change its heuristic value to make everything admissible and consistent.

   Find the range of values that are possible for the selected node.



3. In the design of many real-world socio-technical systems, we often have to design algorithms to perform preference aggregation, the process of aggregating individual inputs (e.g. votes, preferences) into a collective output (e.g. a judgement, winner, or decision).

   Consider a specific instance of the above with an election model. We have a set of n voters in V and a set of m candidates in C, whereby each voter in V has a full strict preference order over all candidates in C. More formally, each voter vi in V has an internal preference ranking of candidates

   $$c_{i,1} > c_{i,2} > \ldots > c_{i,m}$$

   Then, based on these individual preferences, the election committee will select a winner from C via some voting process/algorithm. Note that we assume preferences must be complete in order to derive a winner, and that there will always be a winner for some set of complete voter preferences.

   We would like to simulate the above scenario as a search problem (without some explicit voting process/algorithm), so as to find a set of voter preferences (i.e. the search must consider different permutations of voters' preference ranking), and a corresponding winner (assume that this winner is determined arbitrarily by the committee's voting process; that is, you cannot define your own voting process to determine this winner), such that the following additional properties are satisfied (note that these are applied after a winner is identified by the committee's voting process/algorithm):

   (i) The winner must be ranked at least in the top 50% of any voter's preference list.

   (ii) The winner must be ranked first in at least one voter's preference list.

   (a) (8m) Formulate the above scenario as a search problem by defining the following properties:

      i. state (and initial state)

      ii. actions

      iii. transition model

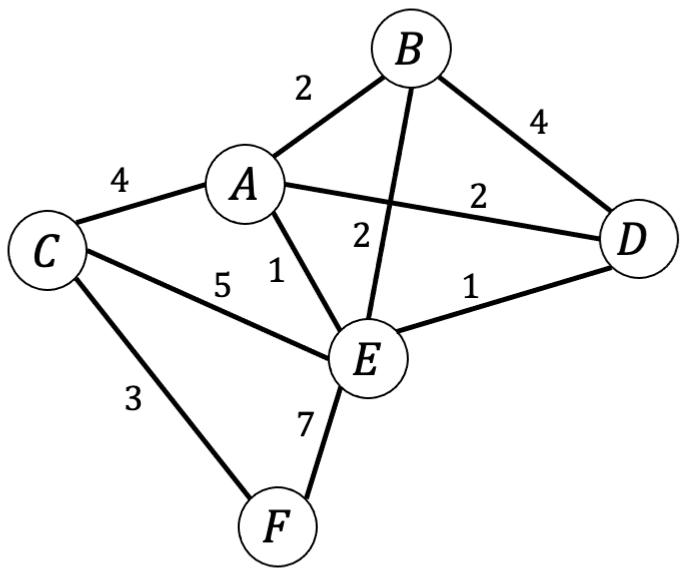
iv. goal test

v. cost function

(b) (2m) What is the size of your state space (in big-O notation)?

(c) (1m) What algorithm would be best suited to solve this problem? Justify.

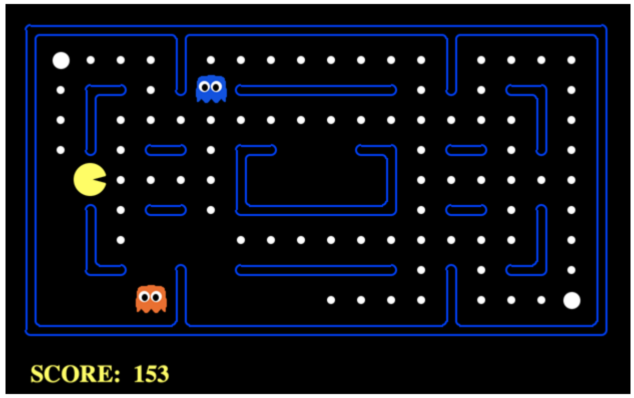
4. Suppose we have the following graph with undirected edges. For BFS & DFS - apply alphabetical tie-breaking rule when pushing into the frontier. For UCS, apply alphabetical tie-breaking when there are multiple nodes of same cost in the PriorityQueue. Perform goal test AFTER popping from the frontier.



Start from the node A. Suppose the goal node is F. Give the sequence of explored (explored means popped from frontier) nodes by tracing each of the following uninformed search algorithms (using graph search):

BFS:
DFS:
UCS:

5. Pac-Man is a maze chase video game where the player controls the eponymous character through an enclosed maze. The objective of the game is to eat all of the dots placed in the maze while avoiding the four ghosts that pursue him. (Source: Wikipedia)



Consider a simplified model of the game, where we exclude the ghosts from the game. Also, the player wins when all of the dots are eaten.

We model the Pac-Man game as such:

- State representation: Position of Pac-Man in the grid at any point of time and the number of pellets left
- Initial State: Grid entirely filled with pellets
- Goal State: No Pellets left in the grid
- Action: Moving up/down/left/right and consuming a pellet
- Transition Model: Eating a pellet at that particular position in the grid.
- Cost function: +1 whenever a move is executed. (Pac-Man move up/down/left/right by 1 unit. Note that consuming a pellet does not incur a cost.)

Consider the following heuristics.

- h_1: Number of pellets left at any point in time.
- h_2: Number of pellets left + minimum of all Manhattan distances from all the pellets to the current Pacman position
- h_3: Maximum value of all the manhattan distances from all the pellets to the current Pacman position.
- h_4: Average Euclidean distance from all pellets to the current Pac-man position

Determine the admissibility of each heuristics. Provide justifications for each of the 4 heuristics.

6. Now compare the dominance of the admissible heuristics that you have selected from the previous question. Justify your answer. *Note that there will be no error carried forward if you have selected any inadmissible heuristic.*

7. Consider the game tree below where the black nodes denote the MAX player and the white nodes denote the MIN player. Both the MAX and MIN players have their own utility scores to compute.

   (a) The maximum possible score for either MAX or MIN player is 20

   (b) MAX player only tries to maximize his own score and does not care what score the MIN player gets.

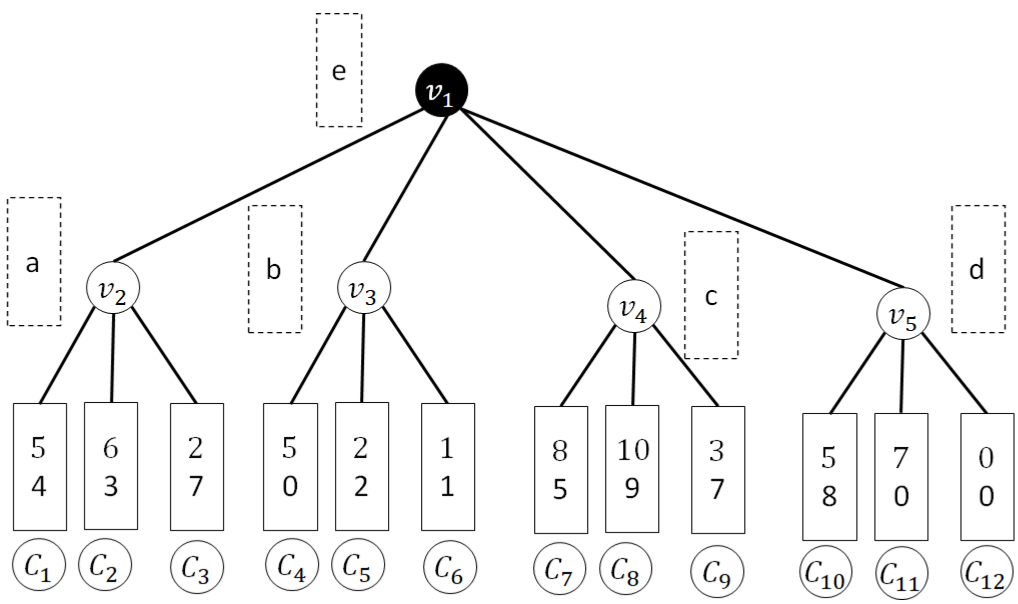
   (c) MIN player is trying to minimize the absolute difference between the two scores. For example, MIN player will prefer a pair of scores (10, 7) as compared to (9, 5) and MIN player prefers a pair of scores (7, 4) as compared to (7, 3).

   (d) In the case of a tiebreaker, MIN player prefers a lower score. For example, MIN player prefers a pair of scores (1, 1) as compared to (10, 10).

   The figure below shows the game tree of MAX node (black) followed by MIN nodes (white). The scores are shown at the leaf nodes with MAX score being on top and the MIN score on the bottom.

   For the next 3 questions, fill in the blanks with the following format: (x, y) (x being the score on top [MAX] and y being the score below [MIN]).

   If the pair of scores is 12 and 6, the correct format will be (12,6). No marks will be awarded if the format given is wrong.



   Using MINIMAX algorithm, determine the pair of scores evaluated by each node. Assume that the order of evaluation is from left to right.

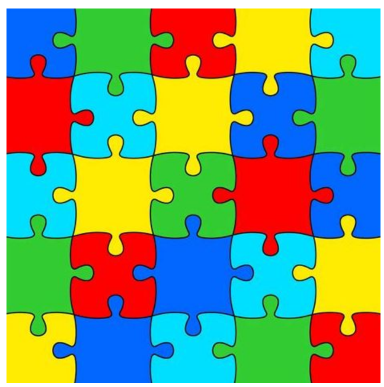
8. Using alpha-beta pruning, determine which leaf nodes are not evaluated. Leaf nodes are labeled as C_1, C_2, ..., C_{12}. Assume that the order of traversal is from left to right.

   Select the leaf nodes that are not evaluated due to pruning.

9. Assume that you now know the order of traversal which maximizes the number of nodes pruned. Using the alpha-beta pruning algorithm with your new knowledge of the optimal ordering, determine which leaf nodes are evaluated.

10. **Extra (Assignment) Question**

    You are given an n-piece unassembled jigsaw puzzle set (you may assume that each jigsaw piece can be properly connected to either 2, 3 or 4 pieces) that assembles into an $(m \times k)$ rectangle (i.e., $n = m \times k$). There may be multiple valid final configurations of the puzzle, the picture illustrates an example.



    Formulate the above as a search problem. More specifically, define the following:

    - State representation
    - Initial state
    - Actions
    - Transition model
    - Step cost
    - Goal test

    If necessary, you may also include assumptions taken here. However, assumptions that are contradictory to any instructions in the question or that are unreasonable will be invalid.