CSE 604: Artificial Intelligence
Quiz 1, July 30, 2018 **Full Marks: 100**
Instructor: Dr. Ahmedul Kabir
Institute of Information Technology
University of Dhaka

Solutions

Problem I. Intelligent Agents [30 Points]

1. Consider an intelligent **chess-playing robot** (a robot**, NOT just a software**). For this robot and its task environment, answer the following questions and justify your answers. [Note: There is not necessarily a "right" answer to each question. The answer can depend on your problem definition and assumptions. So state them clearly].

   a. [10 points] What is meant by **PEAS**? Give a **PEAS description** of the task environment.

      Performance measure: % of wins, playing a decent game of chess, etc.

      Environment: Chess board, pieces, opponent, etc.

      Actuators: Arm, gripper, etc.

      Sensors: Camera, arm joint sensor, etc.

   b. [4 points] Is this a **fully** or **partially observable** environment? Justify your answer.

      Fully observable – since the robot can always see the board with the chess pieces.

      ["**Partially observable**, since the robot cannot see everything in the overall environment" is also a valid answer. However, "It is partially observable because we can't read the opponent's mind" is not a valid reason]

   c. [4 points] Is this a **deterministic** or **stochastic** environment? Justify your answer.

      Deterministic – once the agent decides to move a piece, it knows where it well end up

      [But, since the agent is a robot, mechanical or electrical faults may add an element of uncertainty, in which case it can be classified as **stochastic**. However, "it is stochastic because we don't know what the opponent will do" is not a valid reason]

   d. [4 points] Is this an **episodic** or **sequential** environment? Justify your answer.

      Sequential – Past actions affect future actions. A move is not an isolated event that has nothing to do with the past or future moves.

   e. [4 points] Is this a **simple reflex agent** or a **model based** agent? Justify your answer.

      Model-based agent – the agent needs to build a model of the chess board and pieces and keep updating the model as pieces move. A simple reflex agent would simply look up its expected move from a table of states – which is impossible to maintain for a game of chess.

   f. [4 points] Will this robot pass the **Turing test**? Justify your answer.

      No, the Turing tests requires human-like communication using Natural Language Processing. This robot doesn't have that.

Problem II. Search [24 Points]

In this problem, you are asked to select the best search strategy for each of the situations given below. You can choose any of the informed or uninformed search methods discussed in class. **Justify your answer** briefly but decisively.

a) All the action costs are equal. No heuristic information is given. Time is not a problem, and you have plenty of space. But you must find an optimal solution.

[2 points] Your choice of search method: **Breadth-First Search (BFS)**

[4 points] Justification for choice: If all costs are the same, then a shallowest solution is an optimal solution. Since BFS finds a shallowest solution, then it will be an optimal solution. BFS can take a lot time and space/memory if the search tree is wide and the solution is deep, but these are not issues in this case.

Note: Iterative deepening and uniform search could be also choices for this problem, but there is no need to use them if a more basic algorithm (BFS) can solve the problem. Iterative deepening would take more time (though it would save some space), and uniform search would need to keep track of cost unnecessarily.

b) Cost and heuristic information are given. You need to find an optimal solution and you want to save as much time as possible.

[2 points] Your choice of search method: **A***

[4 points] Justification for choice: Since cost and heuristic information are given, A* will find an optimal solution, assuming that the heuristic information is admissible and consistent. Furthermore, A* is the fasted method that finds an optimal solution.

c) The search space is large and the depth of the solution is not known. No cost and no heuristic information are given. You need to find the shallowest solution but you don't have much memory/space available.

[2 points] Your choice of search method: **Iterative Deepening**

[4 points] Justification for choice: Iterative Deepening will find the shallowest solutions first if the depth limit is increased by 1 at each iteration. Also, Iterative Deeping will save memory/space. [Note that Breadth-First Search would also find the shallowest solution but would take a lot of memory/space, which is an issue in this case.]

d) Heuristic information is given. The search space is finite, but quite large. You want to find a good solution fast but don't have enough memory to keep more than one path at the time.

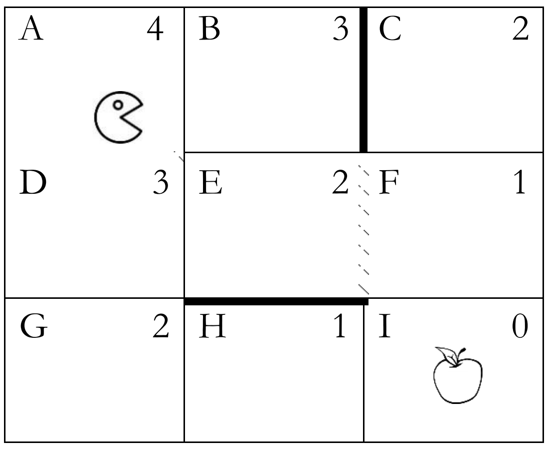
[2 points] Your choice of search method: **Greedy Best First (or A*)**

[4 points] Justification for choice: Since heuristic information is given, we should use either A* or Greedy Best First search. Since there is no mention of an optimal solution, we can possibly find faster solutions using Greedy. However, A* is also acceptable.

Problem III. Informed Search [46 Points]

Consider the very simplified version of Pacman shown below. The game world is a 3 X 3 grid where each cell is named by a letter from A to I. The Pacman agent is initially in cell A, and wants to get to the cell where the apple is so that it can eat the apple. Pacman doesn't know where the apple is, but it is able to detect the presence of the apple when (and only when) it is in the same cell with the apple. Also, Pacman has a heuristic function to estimate how close each cell is to the goal. Here are some more details:

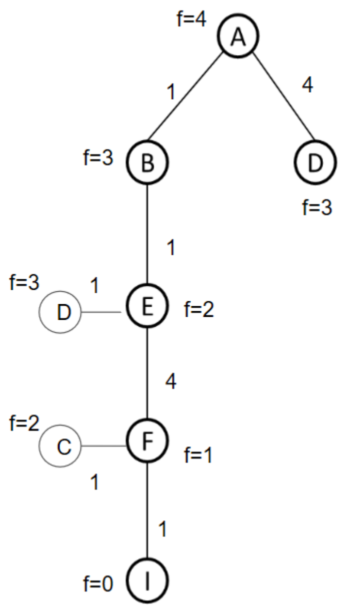
- On the **top right corner of each cell**, the heuristic cost is given (this heuristic function was calculated using the Manhattan distance from each cell to the goal).
- From any cell, the agent can make **horizontal** (left/right) or **vertical** (up/down) moves to an adjacent cell UNLESS the border between the two cells is marked by a **thick line** (borders between B & C and E & H) or if the move would take Pacman outside the grid.
- The **cost** of an allowed move between two cells is 1 (one) EXCEPT for moves between two cells separated by a fence marked with **zigzag lines** (borders between A & D and E & F) for which the move cost is 4 (four).



1. [18 points] A search tree using the **greedy best first search** algorithm is constructed (shown below) from this problem. You need to do the following:
   - [4 points] On the tree, mark each node with the value of the **evaluation function** in that node.
   - [3 points] On the tree, mark each edge with its corresponding **weight** (the cost of that action).
   - [8 points] Simulate the **state of the queue** in each step.
   - [3 points] Calculate the **final cost** to reach the goal.

   Note that there are **no repeated nodes** in a path. Any ties are broken using the **alphabetical order** of the cell names.

   Solution:



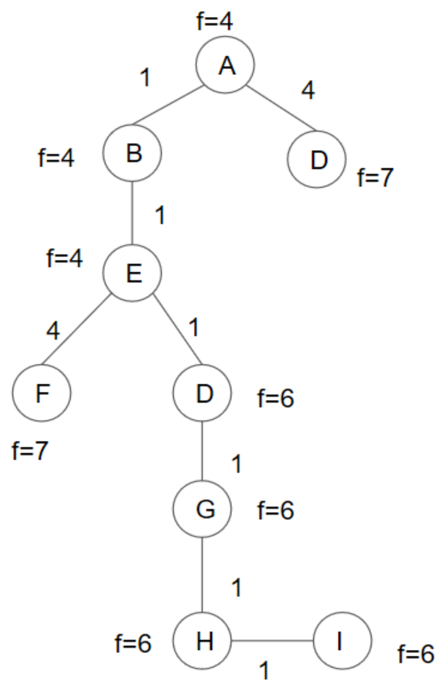
Queue:

1: 4<A>
2: 3<BA>, 3<DA>
3: 2<EBA>, 3<DA>
4: 1<FEBA>, 3<DEBA>, 3<DA>
5: 0<IFEBA>, 2<CFEBA>, 3<DEBA>, 3<DA>

The final cost: 1+1+4+1=7

2. [28 points] Do the same as in the previous page, this time using **A* search** and drawing your own tree. In particular, perform the following tasks.
   - [7 points] Draw the complete **search tree.**
   - [5 points] On the tree, mark each node with the value of the **evaluation function** in that node.
   - [3 points] On the tree, mark each edge with its corresponding **weight** (the cost of that action).
   - [10 points] Simulate the **state of the queue** in each step.
   - [3 points] Calculate the **final cost** to reach the goal.

Solution:



Queue:

1: 4<A>
2: 4<BA>, 7<DA>
3: 4<EBA>, 7<DA>
4: 6<DEBA>, 7<FEBA>, 7<DA>
5: 6<GDEBA>, 7<FEBA>, 7<DA>
6: 6<HGDEBA>, 7<FEBA>, 7<DA>
7: 6<IHGDEBA>, 7<FEBA>, 7<DA>

The final cost: 1+1+1+1+1+1=6