

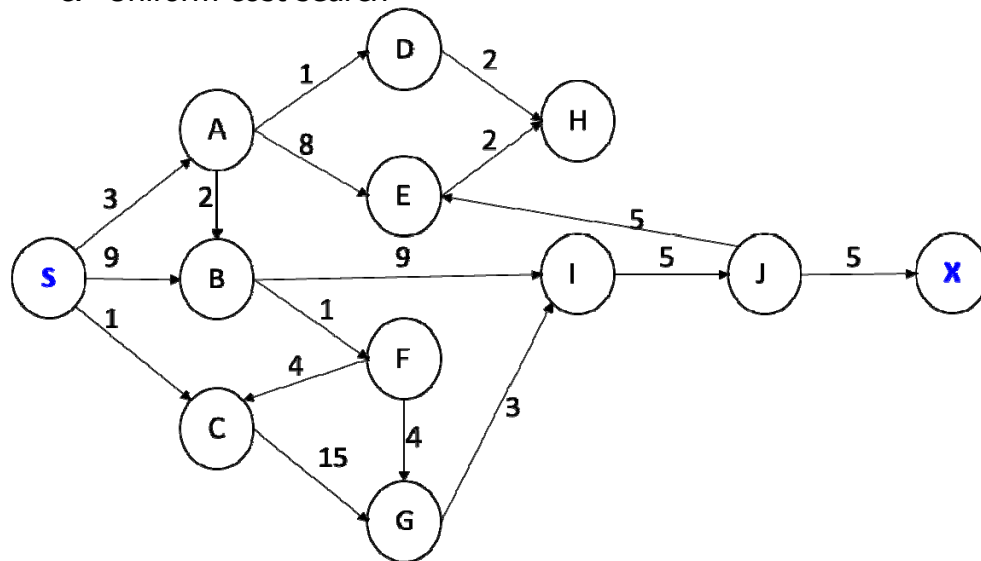
Assignment #1
Artificial Intelligence - CSCE 523
Due: 8:00 AM, Wednesday January 16, 2019
Agents and Uninformed Search

Turnin: E-mail me a zip file containing your typed solution to questions 1 through 5, and your program and report for question 6.

1. (10 points) What is intelligence? There are several research views that can be taken to understand intelligence, for the purpose of this discussion, let us label them philosophical, physicalism, and psychological. Almost every AI algorithm we will discuss is related to one or more of these views. Read the Stanford Encyclopedia of Philosophy articles on the Theory of Mind topics of Qualia and Physicalism (<http://plato.stanford.edu/entries/qualia/> <http://plato.stanford.edu/entries/physicalism/>), and the Sweller paper on Human Cognitive Architecture (http://www.csuchico.edu/~nschwartz/Sweller_2008.pdf) {note: this is the closest I have found to a good short discussion on cognitive psychology and the modeling of it}
 - a. Which view do you feel is more correct? And why?
 - b. Given your choice, is an artificially generated intelligence possible? Why or why not?
2. (15 points) Testing for intelligence: Read Turing's original paper (available on-line at: <http://www.abelard.org/turpap/turpap.htm>). In the paper, he discusses several potential objections to his proposed enterprise and his test for intelligence. Also, refer to the discussion on the Chinese Room Argument found in the text, slides and at: <http://plato.stanford.edu/entries/chinese-room/> or video: <http://www.open.edu/openlearn/history-the-arts/culture/philosophy/60-second-adventures-thought?track=04fb8b569>.
 - a. Which objections still carry some weight? Are his refutations valid?
 - b. Can you think of new objections arising from developments since he wrote the paper?
 - c. Do you think that there is a better test that could be proposed?
3. (5 points) Intelligence and computational limits: There are well-known classes of problems that are intractably difficult for computers and other classes that are provably undecidable by any computer. Does this mean that strong (human level) AI is impossible? Why or why not?
4. (10 points) Consider a modified version of the vacuum-cleaner world (depicted in Figures 2.2 and 2.3 and specified on pages 35-36), in which the geography of the environment – its extent, boundaries, dirt locations,

and obstacles is unknown; the agent can go Up and Down as well as Left and Right.

- a. Can a simple reflex agent be perfectly rational for this environment? Explain.
 - b. Can you design an environment in which your randomized agent will perform very poorly? If not explain, if yes provide an example.
 - c. Can a reflex agent with state outperform a simple reflex agent? Why?
5. (20 points) For the following tree, show the lists of open and visited nodes for each cycle of the listed search algorithms. Expand the nodes in a *right to left* ordering. The start node is **S** and the goal node is **X**. The numbers next to the edges indicate the associated cost. Note: follow the format from class.
- a. Breadth-first search
 - b. Depth-first search
 - c. Uniform cost search



6. (40 points) Ah, Christmas is over. Now, that horrible mall traffic should let up... what is this, a traffic jam?

Seems that everyone is stuck and you can't get through. You must direct the trapped shoppers out of your way and maneuver your trusty vehicle through the maze to get back home in one piece.

For this problem, implement a solver for the Rush Hour puzzle. Rush Hour is a sliding block puzzle containing 4 trucks, 11 cars, and your vehicle placed on a 6x6 grid, with impenetrable walls surrounding its perimeter except for a one cell exit edge. The trucks occupy an area of one by three

cells, and the cars occupy an area of or one by two adjacent grid cells. All vehicles can only move forwards or backwards, never overlapping. The goal is to move the vehicles one at a time so that your car may exit.

The implementation should read a set of puzzles from a text file (format shown in the example below). Where A1 to K1 are cars, O1 to R1 are trucks, and X0 is your vehicle. The exit is always at the third position down on the right, and each empty grid location is (..). The file begins with the number of puzzles found in the file.

Expected turn-in is the 1) source code, 2) compilation and execution instructions, and 3) a short paper describing your solution, results and any search enhancements. A set of 5 problems are provided for testing your search algorithm during development (simple.txt), and 5 that must be tested against in the report (hard.txt), be sure to include your timing and best path results for each of the problems your search can solve. Your implementation must be your work only. As for what enhancements to try – take a look at the publications of Andreas Junghanns on Sokoban (<http://www.cs.ualberta.ca/~games/Sokoban/papers.html>).

To get you started, I have provided Java and MATLAB code that will read the files and handle the state updating (both are in the class directory). For the Java, just implement the search interface in each of your own search(es). For MATLAB, fill in the search function. Note on implementation language – you may find that MATLAB is quicker to write than Java, however you will take a significant performance hit. Using a BFS, the runtime comparison between the implementations on the simple.txt problems are:

Problem	Java	Matlab
1	0.003s	0.03s
2	0.08s	0.13s
3	0.50s	4.30s
4	0.44s	755.89s (13 min)
5	0.92s	(7hr 36min)

```

1
01..Q1Q1Q1B1
01....C1..B1
01X0X0C1.... <= Exit is here, this comment and the arrow
.....P1..... does not appear in the file.
.....P1.....
.....P1.....

```