4511W, Fall-2017
ASSIGNMENT 1 :
Assigned: 09/17/17 Due: 9/24/17 at 11:55 PM (submit via moodle, you may scan your solution for problem 1 if you wish) Submit only pdf or txt files (in a zip if you have multiple files)

Problem 1. (15 points)
For each of the scenarios below, classify the environment based on the seven classifications discussed in class (i.e. fully/partially observable, single/multi-agents, etc.). Additionally for each of the seven classifications, provide a single sentence supporting your reasoning.

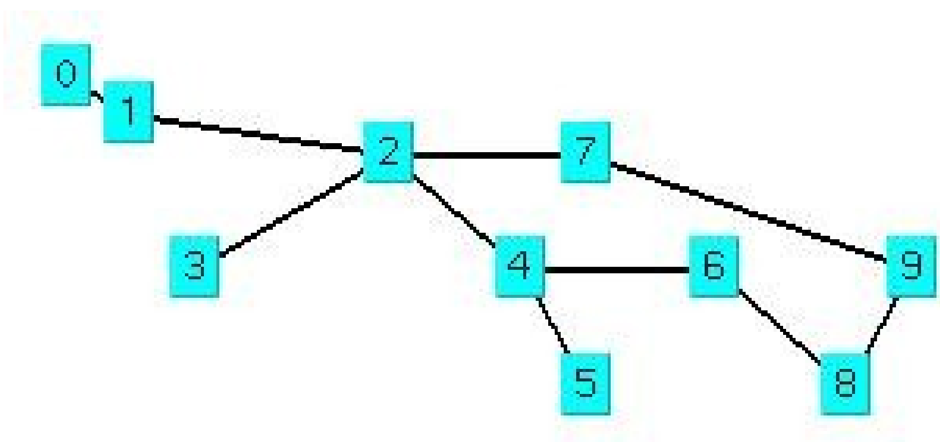
(1) Parking your car in a popular large parking ramp (like at an airport or hospital)

(2) Playing poker ("Draw poker" @ https://en.wikipedia.org/wiki/Poker)

(3) Washing dishes by hand. Assume all the dishes are already in the sink and just need to be scrubbed and moved to a drying rack.

Problem 2. (20 points)
Consider the graph shown below. Assume you start in the square "0" and want to reach the square "8". Show step-by-step how you would solve this with BFS. Use "no-backtracking" version, which keeps track of "explored" nodes. At each step show:
(1) the "fringe"nodes
(2) the "explored" nodes
(3) which node you are taking next from the fringe set to move to the explored set

If you wish to do this pictorially (with colors) rather than using sets, this is acceptable. This picture can be found at: http://www.javacoffeebreak.com/tutorials/aisearch/network.jpg



Problem 3. (15 points)
For each of the situations specify: (a) The initial state, (b) possible actions from the initial state, (c) a general description of other states, and (d) whether the approach is incremental or complete-state.

(1) You just entered a grocery store with a shopping list. Your goal is to reach the checkout with everything on the list.

(2) Suppose we were to implement assigned seating in csci4511. However, we wish to allow preferences, for example friends sitting next to each other.

(3) There is a stack of exams in a random ordering that we wish to sort alphabetically by last-name.

Problem 4. (30 points)
For each of the sub-problems, either give a specific tree/graph for the each scenario or give a clear description of the tree/graph. Note: “finds a goal faster” means removing fewer nodes/states from the fringe set.

(1) Find a tree/graph when “breadth first search” finds a goal faster than “uniform cost search”, but the “breadth first search **is not the optimal** solution.

(2) Find a tree/graph when “breadth first search” finds a goal faster than “uniform cost search”, but the “breadth first search **is the optimal** solution.

(3) Find a tree/graph when “uniform cost search” finds a goal faster than “breadth first search”.

Problem 5. (20 points)
In class I claimed that the worst-case running time of uniform cost search is: $O(b^{1+C^*/min(path\_cost)})$, where b = largest branching factor, C* = path cost from initial to goal, and min(path_cost) = minimum cost of any action. Give an example of a tree/graph that has this worst-case running time.