General Instructions for Labs

Students enrolled in the course are expected to work on programming problems assigned for laboratories in groups. Group once formed would remain fixed for the entire duration of the course. Evaluation of lab assignments should primarily be taken as a two-way feedback, both for students and instructor.

Attending a laboratory session is a must and absence during laboratory sessions without prior intimation to the course instructor/ TA (only genuine cases) will attract 0 credit for that particular session.

Feel free to contact me or the course TA in case you have any query or confusion regarding submissions.

Course Instructor:
Pratik Shah
pratik@iiitvadodara.ac.in

Course TA(s):
Ashish Patel, PhD scholar
201771002@iiitvadodara.ac.in

Priya Sharma, M.Tech. CSE
201961007@iiitvadodara.ac.in

Evaluation Policy

Students are required to work in groups on laboratory problems. Each group is expected to submit a total of two reports containing the discussion on the solution to four problems each. One such report will be due just before the mid-semester examination and the other just before the end-semester examination.

Report/ Term Paper Template : IEEE Transactions on Systems Man and Cybernetics: Systems in LaTeX format.
https://drive.google.com/file/d/1sdEyE-9DxDNFbuR_n3xJ0u0Og6aw9zKU/view?usp=sharing

It is expected that you will write the report as if you are writing a tutorial to explain the problems and the corresponding solutions.

A viva voce will be conducted immediately before/ after the end-semester examination.

Moral Code of Conduct

Zero tolerance to plagiarism. If the report submitted by any group is found to have copied extracts from any of the other submitted reports the group members will receive no credit for the entire report. In a report if some material is found to have been reproduced from some source without proper citation or reference, the report will attract zero credits.

Week 0 : 4 Jan - 8 Jan 2021

Lab Assignment - 0

Learning Objective: To be able to model a given problem in terms of state space search problem and solve the same using BFS/ DFS

Reference:

[1] Artificial Intelligence: a Modern Approach, Russell and Norvig (Fourth edition)
Chapter 1, 2, 3

Problem Statement:

A. In the rabbit leap problem, three east-bound rabbits stand in a line blocked by three west-bound rabbits. They are crossing a stream with stones placed in the east west direction in a line. There is one empty stone between them. The rabbits can only move forward one step or two steps. They can jump over one rabbit if the need arises, but not more than that. Are they smart enough to cross each other without having to step into the water?
B. The missionaries and cannibals problem is usually stated as follows. Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place. This problem is famous in AI because it was the subject of the first paper that approached problem-formulation from an analytical viewpoint.

For the above two problems,

1. Model the problem as a state space search problem. How large is the search space?
2. Solve the problem using BFS. The optimal solution is the one with the fewest number of steps. Is the solution that you have acquired an optimal one? The program should print out the solution by listing a sequence of steps needed to reach the goal state from the initial state.
3. Solve the problem using DFS. The program should print out the solution by listing a sequence of steps needed to reach the goal state from the initial state
4. Compare solutions found from BFS and DFS. Comment on solutions. Also compare the time and space complexities of both.

Week 1 : 11 Jan - 15 Jan 2021

Lab Assignment - 1

Learning Objective: To design a graph search agent and understand the use of a hash table, queue in state space search.

Reference:

[1] Artificial Intelligence: a Modern Approach, Russell and Norvig (Fourth edition)
Chapter 2 and 3

[2] A first course in Artificial Intelligence, Deepak Khemani
Chapter 2

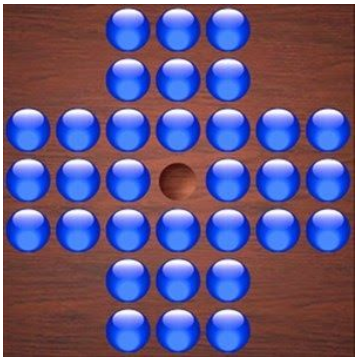
Problem Statement:

A. Write a pseudocode for a graph search agent. Represent the agent in the form of a flow chart. Clearly mention all the implementation details with reasons.
B. Write a collection of functions imitating the environment for Puzzle-8.
C. Describe what is Iterative Deepening Search.
D. Considering the cost associated with every move to be the same (uniform cost), write a function which can backtrack and produce the path taken to reach the goal state from the source/ initial state.
E. Generate Puzzle-8 instances with the goal state at depth “d”.
F. Prepare a table indicating the memory and time requirements to solve Puzzle-8 instances (depth “d”) using your graph search agent.

Week 2 : 18 Jan - 22 Jan 2021

Lab Assignment - 2

Learning Objective: To understand the use of Heuristic function for reducing the size of the search space. Explore non-classical search algorithms for large problems.



Reference:

[1] Artificial Intelligence: a Modern Approach, Russell and Norvig (Fourth edition)
Chapter 3 and 4

[2] A first course in Artificial Intelligence, Deepak Khemani
Chapter 3, 4

Problem Statement:

A. Read about the game of marble solitaire. Figure shows the initial board configuration. The goal is to reach the board configuration where only one marble is left at the centre. To solve marble solitaire, (1) Implement priority queue based search considering path cost, (2) suggest two different heuristic functions with justification, (3) Implement best first search algorithm, (4) Implement A*, (5) Compare the results of various search algorithms.
B. Write a program to randomly generate k-SAT problems. The program must accept values for k, m the number of clauses in the formula, and n the number of variables. Each clause of length k must contain distinct variables or their negation. Instances generated by this algorithm belong to fixed clause length models of SAT and are known as uniform random k-SAT problems.
C. Write programs to solve a set of uniform random 3-SAT problems for different combinations of m and n, and compare their performance. Try the Hill-Climbing, Beam-Search with beam widths 3 and 4, Variable-Neighborhood-Descent with 3 neighborhood functions. Use two different heuristic functions and compare them with respect to *penetrance*.

Week 3 : 18-22 January, 2021

Lab Assignment 3

Learning Objective:

Non-deterministic Search | Simulated Annealing

For problems with large search spaces, randomized search becomes a meaningful option given partial/ full-information about the domain.

Reference:

[1] A first course in Artificial Intelligence, Deepak Khemani (Chapter 4)

Problem:

Travelling Salesman Problem (TSP) is a hard problem, and is simple to state. Given a graph in which the nodes are locations of cities, and edges are labelled with the cost of travelling between cities, find a cycle containing each city exactly once, such that the total cost of the tour is as low as possible.

For the state of Rajasthan, find out at least twenty important tourist locations. Suppose your relatives are about to visit you next week. Use Simulated Annealing to plan a cost effective tour of Rajasthan. It is reasonable to assume that the cost of travelling between two locations is proportional to the distance between them.

An interesting problem domain with TSP instances:

VLSI: http://www.math.uwaterloo.ca/tsp/vlsi/index.html#XQF131

(Attempt at least five problems from the above list and compare your results.)

Week 4 : 25-29 January, 2021

Week 1-3 : Challenge Problem

Learning Objective:
Non-deterministic Search | Simulated Annealing | Genetic Algorithm
For problems with large search spaces, randomized search becomes a meaningful option given partial/ full-information about the domain.

Reference:
[1] Simulated annealing from basics to applications, Daniel Delahaye, Supatcha Chaimatanan, Marcel Mongeau
[2] A first course in Artificial Intelligence, Deepak Khemani (Chapter 4)
[3] Artificial Intelligence: a Modern Approach, Russell and Norvig (Fourth edition)
Chapter 3 and 4

Problem:

Ever played jigsaw puzzles in childhood? Time to make an agent learn to solve the puzzle.

Some rough, confusing code snippets are posted on the codes folder of the course along with the scrambled_lena.mat

Formulate the problem as a state space search problem. Try to solve the same using simulated annealing.

Have fun!

This is a bonus challenge! If you solve this, you are up for some extra credits.

Week 5 : 1-5 February, 2021

Lab Assignment 4

Learning Objective:
Game Playing Agent | Minimax | Alpha-Beta Pruning

Systematic adversarial search can lead to savings in terms of pruning of sub-trees resulting in lesser node evaluations

References:
[1] A first course in Artificial Intelligence, Deepak Khemani (Chapter 8)
[2] Artificial Intelligence: a Modern Approach, Russell and Norvig (Fourth edition) (Chapter 5)

Problem:

1. What is the size of the game tree for Noughts and Crosses? Sketch the game tree.
2. Read about the game of Nim (a player left with no move losing the game). For the initial configuration of the game with three piles of objects as shown in Figure, show that regardless of the strategy of player-1, player-2 will always win. Try to explain the reason with the MINIMAX value backup argument on the game tree.
3. Implement MINIMAX and alpha-beta pruning agents. Report on number of evaluated nodes for Noughts and Crosses game tree.
4. Using recurrence relation show that under perfect ordering of leaf nodes, the alpha-beta pruning time complexity is $O(b^{m/2})$, where b is the effective branching factor and m is the depth of the tree.

Week 5 : 8-12 February, 2021

Lab Assignment 5

Learning Objective:

Understand the graphical models for inference under uncertainty, build Bayesian Network in R, Learn the structure and CPTs from Data, naive Bayes classification with dependency between features.

Reference:

1. https://www.bnlearn.com/
2. http://gauss.inf.um.es/umur/xjurponencias/talleres/J3.pdf

Problem Statement:

A table containing grades earned by students in respective courses is made available to you in (laboratory_work folder) 2020_bn_nb_data.txt.

1. Consider grades earned in each of the courses as random variables and learn the dependencies between courses.
2. Using the data, learn the CPTs for each course node.
3. What grade will a student get in PH100 if he earns DD in EC100, CC in IT101 and CD in MA101.
4. The last column in the data file indicates whether a student qualifies for an internship program or not. From the given data, take 70 percent data for training and build a naive Bayes classifier (considering that the grades earned in different courses are independent of each other) which takes in the student's performance and returns the qualification status with a probability. Test your classifier on the remaining 30 percent data. Repeat this experiment for 20 random selection of training and testing data. Report results about the accuracy of your classifier.
5. Repeat 4, considering that the grades earned in different courses may be dependent.