19th December, 2016

COMPUTER SCIENCE LAB (C-IV): Discrete Structures Practical Lab

Practical: 60 Lectures
(4 Practicals per week (each in group of 10 to 15))

Note: From a computer science perspective, for better understanding of important discrete structures topics like Graphs, students must be taught from an implementation point of view. For the programs on recursion, students must print the intermediate steps, if possible. The computer language used for implementation would be C++.

1. Write a Program containing following set operation functions for a given set A (consider A to be array of integers).
   a) unique (A) ; return a set of the unique elements of a given set.
      (For example unique ([1, 2, 8, 2, 2, 1, 7, 2, 1, 1]) should return 1,2,8,7)
   b) ismember (a, A); for checking an element's membership, with return value as true/false
   c) cardinality(A): to determine the cardinality of a set;
   d) powerset(A): to list all the elements of power set of A
2. Create a class SET and include functions to perform following Set operations on Sets: Subset, Union, Intersection, Complement, Set Difference, Symmetric Difference and Cartesian Product. Write a Program which takes sets from user and use this class.
3. Create a class RELATION, use Matrix notation to represent a relation. Include functions to check if a relation is reflexive, Symmetric, Anti-symmetric, Transitive, to generate Reflexive Closure of relation. Write a Program to use this class.
4. Use the functions defined in Ques 3 to find check whether the given relation is:
   i) Equivalent, or
   ii) Partial Order relation, or
   iii) None
5. Write a Program to generate the Fibonacci Series using recursion.
6. Write a Program to implement Tower of Hanoi using recursion.
7. Write a Program to implement binary search using recursion.
8. Write a Program to implement Bubble Sort. Find the number of comparisons during each pass and display the intermediate result as well.
9. Write a Program to implement Insertion Sort. Find the number of comparisons during each pass and display the intermediate result as well.
10. Write a Program that generates all the permutations of a given set of digits, with or without repetition. (For example, if the given set is {1,2}, the permutations are 12 and 21). (One method is given in Liu)
11. Write a Program to calculate Permutation and Combination (${}^{n}C_{r}$ and ${}^{n}P_{r}$) using recursion.
12. For any number **n**, write a program to list all the solutions of the equation $x_1 + x_2 + x_3 + \ldots + x_n = C$, where C is a constant, where $x_1, x_2, x_3, \ldots, x_n$ are nonnegative integers.

13. Write a Program to accept the truth values of variables **x** and **y**, write a program to print the truth table of the following statements:
    a) Conjunction
    b) Disjunction
    c) Exclusive OR
    d) Conditional
    e) Bi-conditional
    f) Exclusive NOR
    g) Negation
    h) NAND
    i) NOR
14. Write a Program to solve second order linear homogeneous recurrence relation with constant coefficients.
15. Write a program to accept an input **n** from the user, Tabulate the values of **T(n)** where **n** varies from **0** to **n** for the recurrence relations. For e.g. T(n) = T(n-1) + n, T(0) = 1, T(n) = T(n-1) + n^2, T(0) =1, T(n) = 2*T(n)/2 + n, T(1)=1.
16. Write a Program to store a function (polynomial/exponential) in an array, and then evaluate the polynomial. (For example store **f(x) = 4n^3 + 2n + 9** in an array and for a given value of **n**, say **n = 5**, evaluate (i.e. compute the value of **f(5)**).
17. Write a Program to represent Graphs using the Adjacency Matrices and check if it is a complete graph.
18. Write a Program to accept a directed graph G and compute the in-degree and out-degree of each vertex.
19. Given a graph G, Write a Program to find the number of paths of length **n** between the source and destination entered by the user.
20. Given an adjacency matrix of a graph, write a program to find out if there exists Euler path / Euler Circuit.
21. Given a full **m-ary** tree with **i** internal vertices, Write a Program to find the number of leaf nodes.