CS2206 ALGORITHMS LAB Through C++

Course objectives:

- The laboratory component will emphasize two areas:
- Implementation of algorithms covered in class: This will involve running the algorithms under varying input sets and measuring running times, use of different data structures for the same algorithm (wherever applicable) to see its effect on time and space, comparison of different algorithms for the same problem etc.
- Design of Algorithms: This will involve design and implementation of algorithms for problems not covered in class but related to topics covered in class.
- The exact set of algorithms to design and implement is to be decided by the instructor. In addition, there will be at least one significantly large design project involving some real world application. An efficient design of the project should require the use of multiple data structures and a combination of different algorithms/techniques.

Course Outcomes:

The student should be able to:

- Design algorithms using appropriate design techniques (brute-force, greedy, dynamic programming, etc.)
- Implement a variety of algorithms such assorting, graph related, combinatorial, etc., in a high level language.
- Analyze and compare the performance of algorithms using language features.
- Apply and implement learned algorithm design techniques and data structures to solve real-world problems.

Programs List:

1. a) Create a CPP class called Student with the following details as variables within it.

   (i) Register_number

   (ii) Student_name

   (iii) Programme_name

   (iv) Phone _number

   Write a program to create nStudent objects and print the Register_number, Student_name, Programme_name, and Phone_number of these objects with suitable headings.

   b). Write a program to implement the Stack using arrays. Write Push (), Pop(), and Display() methods to demonstrate its working.

2. a). Design a superclass called Staff with details as StaffId, Name, Phone, Salary. Extend this class by writing three subclasses namely Teaching (domain, publications), Technical (skills), and Contract (period). Write a CPP program to read and display at least 3 staff objects of all three categories.

b). Write a class called Customer to store their name and date_of_birth. The date_of_birth format should be dd/mm/yyyy. Write methods to read customer data as and display as using StringTokenizer class considering the delimiter character as "/".

3. a). Write a program to read two integers a andb. Compute a/b and print, when b is not zero. Raise an exception when b is equal to zero.

b). Write a program that implements a multi-thread application that has three threads. First thread generates a random integer for every 1 second; second thread computes the square of the number and prints; third thread will print the value of cube of the number.

4. Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using CPP how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.

5. Sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of n> 5000, and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using CPP how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.

6. Implement the Knapsack problem using (a) Dynamic Programming method (b) Greedy method.

7. Write a program-from a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm..

8. Find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm. Use Union-Find algorithms in your program.

9. Find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.

10. Write programs to

(a) Implement All-Pairs Shortest Paths problem using Floyd's algorithm.

(b) Implement Travelling Sales Person problem using Dynamic programming.

11. Design and implement in CPP, to find a subset of a given set S = {Sl, S2,.....,Sn} of n positive integers whose SUM is equal to a given positive integer d. For example, if S ={1, 2, 5, 6, 8} and d= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution.

12. Design and implement in CPP to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using backtracking principle.

REFERENCES:

1. T. H. Cormen, C. L. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, MIT Press.
2. J. Kleinberg and E. Tardos, Algorithm Design, Addison-Wesley.
3. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins.
4. A. Gibbons, Algorithmic Graph Theory, Cambridge University Press.
5. Michael T. Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, John Wiley.
6. R. Sedgewick, Algorithms in C (Parts 1-5), Addison Wesley.
7. M. H. Alsuwaiyel, Algorithm Design Techniques and Analysis, World Scientific.
8. Gilles Brassard and Paul Bratley, Algorithmics: theory and practice, Prentice-Hall.
9. Udi Manber, Introduction to Algorithms: A Creative Approach, Addison-Wesley.
10. Sara Baase and Allen Van Gelder, Computer Algorithms: Introduction to Design and Analysis, Addison-Wesley.