COMP 314
ALGORITHMS AND COMPLEXITY
3 CREDIT

F.M.: 100

Total Hours: 48 [Theory + Practical]

Course Objectives:

- Develop a broad understanding of standard algorithms and their common uses
- Be able to analyze the asymptotic performance of a variety of algorithms
- Be able to experimentally test the performance of a particular algorithm in a particular context
- Develop a degree of fluency in the mathematical techniques used to demonstrate correctness
- Develop and implement algorithms needed in disciplined problem solving
- Develop an understanding of NP-complete (hard) problems and approximation algorithms

Prerequisites:

It is expected that students have been introduced to some prior courses on programming and data structures. For the understanding and implementation of algorithms, it is essential that students have a fairly good command of some high level programming languages like C, C++ or Java.

Contents:

1. **Introduction to algorithms:** [9 Hrs.]

   1.1 Mathematical preliminaries of foundations:

      1.1.1 Growth of functions,
      1.1.2 Summations,
      1.1.3 Recurrences

   1.2 Analysis of sorting algorithms

      1.2.1 Selection sort
      1.2.2 Insertion sort
      1.2.3 Merge sort
      1.2.4 Quick sort
      1.2.5 Heap sort

2. **Data structures revisited:** [7 Hrs.]

   2.1 Stack,
   2.2 Queue,
   2.3 Linked list,
   2.4 Hash tables,
   2.5 Binary search trees,
   2.6 Red-Black trees

3. **Algorithmic Strategies:** [10 Hrs.]

   3.1 Brute-force algorithms,
   3.2 Greedy algorithms: action-selection problem and Huffman codes,
   3.3 Divide and Conquer,
   3.4 Backtracking,
   3.5 Branch-and-bound

4. Dynamic Programming [6 Hrs.]

4.1 Matrix chain multiplication method
4.2 Longest common subsequence

5. Graph Algorithms [7 Hrs.]

5.1 Representation,
5.2 Traversal techniques,
5.3 Minimum spanning tree,
5.4 Shortest path algorithms, Dijkstra's algorithm,
5.5 Flow networks

6. NP-Completeness [6 Hrs.]

6.1 NP-completeness and the classes P and NP,
6.2 Polynomial time verification,
6.3 NP-completeness and reducibility,
6.4 NP-complete problems.

TEXT BOOKS:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. Introduction to Algorithms. Third Edition.
2. Horowitz E., Sahni S., and Rajasekaran S. Fundamentals of Computer Algorithms, Second Edition.
3. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Data Structures & Algorithms in Python, Wiley & Sons, 2013.
4. Rance D. Necaise, Data Structures and Algorithms Using Python, John Wiley & Sons, Inc., 2011.
5. Problem Solving with Algorithms and Data Structures – interactive book content available at www.interactivepython.org

REFERENCES:

1. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman. The Design and Analysis of Computer Algorithms. Fourth Indian Reprint, 2001.
2. S.E. Goodman, S.T. Hedetniemi. Introduction to the Design and Analysis of Algorithms. Fifth Printing, 1988.
3. Sartaj Sahni. Data Structures, Algorithms and Applications in Java. Second Edition.
4. Horowitz E., Sahni S. and Anderson-Freed S. Fundamentals of Data Structures in C. Second Edition.

EVALUATION:

Internal: 50

- Lab and Practical assignments: 15
- First Internal: 10
- Second Internal: 10
- Quiz: 10
- Final viva: 5

External: 50