Department Of Computer Science and Engineering
Kathmandu University
Dhulikhel, Kavre

Subject: Algorithms and Complexity **Course Code: COMP 314**
Level: B.E./B.Sc 3rd Year/2nd Semester **Credit Hours: 3**

Course Description

This course will introduce students to various algorithm design techniques, and some intermediate and advanced data structures. It covers some standard algorithms to facilitate students in understanding algorithm design strategies and algorithm complexity analysis. It also introduces students to computational complexity issues.

Course Objectives

- Develop a broad understanding of standard algorithm design strategies
- Be familiar with intermediate and advanced data structures 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 (e.g. COMP 202 course). 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.

CHAPTERS

1. **Introduction to algorithms** [8 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. **Some data structures** [10 Hrs.]
1. Data Structures Basics: Stack, Queue, Linked List, Binary Search Tree.
2. Efficient binary search trees - AVL tree, Red-Black trees
3. Multiway search trees - B-tree, B+-tree
4. Priority queues - Binomial heap, Fibonacci heap
5. Graph data structure - representation, traversal

2. Algorithmic Strategies [12 Hrs.]
2.1. Brute-force algorithms,
2.2. Greedy algorithms:
2.2.1. Action-selection problem
2.2.2. Huffman coding
2.2.3. Minimum spanning tree algorithms - Kruskal's, Prim's
2.2.4. Shortest path algorithm - Dijkstra's
2.2.5. Flow networks - Ford Fulkerson algorithm
2.3. Divide and Conquer,
2.4. Backtracking - N-queen problem
2.5. Branch-and-bound - 0/1 Knapsack problem

3. Dynamic Programming [4 Hrs.]
3.1. Matrix chain multiplication method
3.2. Longest common subsequence

4. Probabilistic and Parallel algorithms [8 Hrs.]
4.1. **Probabilistic algorithms**
4.1.1. Monte Carlo algorithm - Primality testing
4.1.2. Las Vegas algorithm - N-queen problem

4.2. **Parallel algorithms**

4.2.1. Finding connected components in a graph

4.2.2. Parallel sorting

5. NP-Completeness [3 Hrs.]

5.1. NP-completeness and the classes P and NP,

5.2. Polynomial time verification,

5.3. NP-completeness and reducibility,

5.4. NP-complete problems.

Text Books

1. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to algorithms. MIT press, 2009.
2. Horowitz, Ellis, Sahni Sartaj, and Rajasekaran Sanguthevar. Fundamentals of Computer Algorithms, Second Edition.
3. Goodrich, Michael T., Roberto Tamassia, and Michael H. Goldwasser. Data structures and algorithms in Python. John Wiley & Sons Ltd, 2013.
4. Necaise, Rance D. Data Structures and Algorithms Using Python. Wiley Publishing, 2010.

Reference Books

1. Aho, Alfred V., and John E. Hopcroft. The design and analysis of computer algorithms, Fourth Indian Reprint. Pearson Education India, 2001.
2. Goodman, Seymour E., and Stephen T. Hedetniemi. Introduction to the Design and Analysis of Algorithms. Fifth Printing, 1988.
3. Sahni, Sartaj. Data Structures, Algorithms and Applications in Java. Second Edition.
4. Horowitz, Ellis, Sartaj Sahni, and Susan Anderson-Freed. Fundamentals of data structures in C. Second Edition.