University of Tabuk
College of Computer and Information Technology
Department of Computer Science

Syllabus

Course

Code: *CSC321*

Title: ***Design and Analysis of Algorithms***

Credits

3-0-2-3

Text Books

- Levitin, Anany. *Introduction to the Design & Analysis of Algorithms, 2nd Edition*. Addison Wesley, 2007. ISBN 0-321-35828-7.

References

- Steven S. Skiena The Algorithm Design Manual Second Edition, Springer-Verlag London Limited 2008, ISBN: 978-1-84800-069-8 e-ISBN: 978-1-84800-070-4
- Michael Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, John Wiley, 2002

Prerequisite:

CSC220

Course Description

The objective of this course is to teach students methods of designing and analyzing algorithms. Specific topics include computing worst case and average case complexity for an algorithm, design techniques, dynamic programming, divide-and-conquer, greedy method, branch-and-bound, backtracking, graph traversals, minimum spanning tree, and shortest path algorithms.

Objectives:

- Prove the correctness of simple algorithms
- Understand asymptotic notation
- Analyze simple iterative and recursive algorithms
- Use the divide-and-conquer, greedy, and dynamic programming paradigms to design algorithms

- Learn a variety of useful algorithms.

Course Outline

Week	**Topics**
1	Introduction to algorithms, Logarithms,
2	Analysis framework: *O*, Θ, Ω notations, Calculating worst and average complexity
3	Mathematical analysis: non-recursive and recursive algorithms
4	Brute-force algorithms Exhaustive search
5	Divide-and-conquer: Mergesort, Quicksort
6	Binary search, Closest Pair/Convex Hull
7	Data structures for graphs, Breadth-first search
8	Topological sort/connectivity
9	Minimum spanning trees, Shortest paths
10	Instance simplification: Presorting, Gaussian Elimination, Balanced search trees
11	Space/time tradeoffs: String Matching
12	Dynamic Programming: Binomial Coefficient, Warshall’s and Floyd’s Algorithms, Knapsack Problem
13	Greedy Algorithms: Prim’s and Kruskal’s Algorithms,
14	Dijkstra’s Algorithm, Huffman Trees
15	The NP-completeness challenge, Approximation Algorithms
16	Final Exam

Grading

Assessment/Evaluation:

Midterm 1	**20%**
Midterm 2	**20%**
Assignments	**5%**
Quizzes	**5%**

Project	**10%**
Final Exam	**40%**
Total	**(100%)**

Intended Learning Outcomes:

Upon completion, students will be able to:

Outcomes	Assessment Methods
• Understand and use fundamental algorithms and algorithmic techniques.	Exams Quizez
• Decide which algorithm among a set of possible choices is best for a given application.	Exams Quizez
• Prove correctness and analyze the running time of a given algorithm.	Exams Quizez

Method of Teaching:

- **Lectures (three hours per week)**
- **Tutorial (two hours per week)**