Course Syllabus

1. Course Title — Design and Analysis of Algorithms
2. Course Credit — 3
3. Faculty — Science, Department of Mathematics
4. Semester — 1st Semester
5. Instructor — Nagul Cooharojananone
6. Course Condition — No
7. Course Status — Subject
8. Program name — Master degree of Computer Science
9. Course Level — Master degree
10. Hours per a week — Lecture 3 hours
11. Course Description

    Analysis of Algorithm, Brute Force, Divide and Conquer, Decrease and Conquer, Transform and Conquer, Dynamic Programming, Greedy Technique, P and NP Problems

12. Course Outline

    12.1 Purpose

    1. Students are able to understand the fundamentals of Algorithmic Problem Solving and Problem Types.
    2. Students are able to understand the fundamentals of the Analysis of Algorithm Efficiency.
    3. Students are able to solve problems using Brute Force, Divide and Conquer and Dynamic Programming.
    4. Students are able to understand the P and NP problems.
    5. Students are able to understand the fundamental of Data Structures

    12.2 Outline (See a table)

    12.3 Teaching Method

    1. Lecture
    2. Examination
    3. Project

    12.4 Teaching Media

    1. White board
    2. Overhead projector
    3. Computer

    12.5 Evaluation: 100 points full score is classified into

    1. Exercises — 30 points
    2. Midterm — 35 points
    3. Final — 35 points

13. Reference Books

1. Anany Levitin, Introduction to the Design and Analysis of Algorithms, Pearson Education.
2. Michael T. Goodrich and Roberto Tamassia, Algorithm Design Foundations, Analysis, and Internet Examples, John Wiley & Sons, Inc., 2001.

3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, MIT Press, 2001

2301681 Design and Analysis of Algorithms

1st Semester, Year 2552

Midterm: 20 July 2552 Time: 18:00 - 21:00 **Final:** 21 September 2552 Time: 18:00 - 21:00

Week / **Chapter**	**Topics**
Week 1 - Introduction - Analysis of algorithms	- What is an algorithm? - Why do we need to concern about algorithm? - Fundamentals of Algorithmic Problem Solving - Important Problem Types - Problems and instances
Week 2 - Fundamental data structures - Analysis of algorithms	- Linear data structures - Graphs - Trees - Measuring the efficiency of algorithms - Orders of Growth
Week 3 - Analysis of algorithms	- Worse-Case, Best-Case Analysis - Asymptotic notation - Useful property involving the Asymptotic Notations - Using limits for comparing orders of growth
Week 4 - Analysis of algorithms	- Mathematical analysis of nonrecursive algorithms - Mathematical analysis of recursive algorithms - Master Theorem
Week 5 - Brute-Force Algorithms	- General characteristics of brute-force algorithms - Selection sort and bubble sort - Sequential search - Brute-Force string matching
Week 6 - Brute-Force Algorithms - Divide and Conquer	- Closest-Pair and Convex-Hull problems by Brute Force - Exhaustive Search - Mergesort - Quicksort
Week 7 - Divide and Conquer	- Binary search - Binary tree traversals and related properties - Multiplication of large integers - Strassen's Matrix multiplication - Convex hull

Week 8 - Decrease and Conquer	- Insertion sort - Depth-First search and Breath-First search - Topological Sorting - Decrease by a constant factor algorithms - Variable size decrease algorithms
Week 9 - Transform and Conquer	- Presorting - Balanced search trees - Heaps and Heap sort - Horner's rule and binary Exponentiation
Week 10 - Space and Time Tradeoffs	- Sorting by Counting - String Matching - Hashing - B-trees
Week 11 - Dynamic Programming	- Principle of Optimality - Computing a binomial coefficient - Warshall's and Floyd's algorithms
Week 12 - Dynamic Programming - Greedy Technique	- Optimal binary search trees - The knapsack problem and memory functions - Prim's Algorithm
Week 13 - Greedy Technique	- Kruskal's Algorithm - Dijkstra's Algorithm - Huffman Trees
Week 14 - Iterative Improvement	- The simplex method - The maximum flow problem - Maximum matching in bi partite graphs
Week 15 - Limitations of Algorithm Power	- NP-Hard problems - NP-complete problems - Other complexity classes