**FACULTY OF INFORMATION TECHNOLOGY**

**CURRICULUM SYLLABUS**

|  |  |
| --- | --- |
| **COURSE** | **KI141319 : Design and Analysis of Algorithms 2** |
|  | Credit : 3 |
|  | Semester : 4 |

|  |  |
| --- | --- |
| **COURSE DESCRIPTION** | |
| In this course students will learn about optimal abstraction from real problems with medium and hard complexity level. Student are also able to implements the abstraction into design of algorithms with regards to correctness and efficiency using formal representation. In the end, students also able to present whole steps in design and analysis of algorithms systematically, both written and oral. | |
| **LEARNING ACHIEVEMENT OF SUPPORTED STUDY PROGRAM** | |
|  | |
| 1.1.2 | Able to identify, analyze, and solve problems systematically and organized in information technology and communication |
| 1.1.3 | Able to use and implements theoretical and empirical concepts in problem solving in information technology and communication |
| 3.1.1 | Able to analyze the resource requirements in problem solving in information technology and communication |
| 3.2.3 | Have creativity in developing various solutions |
| 3.2.4 | Able to communicate, both oral and written |
| 3.3.1 | Responsible with their own results |
|  | |
| **COURSE OUTCOMES** | |
| * Students able to make optimal abstraction creatively from real problems with medium and hard level complexity. * Students able to implement problems abstraction into design of algorithms with regards to correctness and efficiency. * Students able to express correctness analysis and complexity of design of algorithm formally. * Students able to implement design of algorithms result into programming language and test the accuracy in online judge * Students able to represent whole steps in design of algorithms systematically, both oral and written. | |
| **DISCUSSION SUBJECTS** | |
| * Algorithm and complexity * Design and analysis of algorithm with divide and conquer paradigm   + Binary search algorithm   + Non-classical dynamic programming   + Greedy algorithm * Representation of several advance data structures that related to dynamic programming   + Tree segment structure (range min/max query, range sum query) and lazy propagation   + Fenwick Tree (binary indexed tree)   + Splay tree * Design and analysis of algorithms in graph structures   + Minimum spanning tree   + All pair shortest path and single source shortest path   + Strongly connected component, topological sort and 2-SAT problem   + Maximum flow, minimum cut, and bipartite matching * Design and analysis of algorithms in string matching problem   + KMP, Boyer Moore   + Trie data structure   + Suffix array, suffix tree and suffix trie | |
| **PREREQUISITE** | |
| Design and Analysis of Algorithms 1 | |
| **PRIMARY REFERENCES** | |
| 1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, “Introduction to Algorithms Third Edition”, MIT Press, 2009 | |
| **SUPPORTED REFERENCES** | |
| 1. Levitin, Anany, “Introduction to The Design & Analysis Af algorithms 3rd ed”, Addison-Wesley, 2012 2. Robert Sedgewick, Kevin Wayne, Algorithms, 4th Edition, Addison Wesley, 2011 3. Stephen Halim, Felix Halim, Competitive Programming, 3rd Edition, NUS School of Computing, 2013 | |