Course Outline

COMP6049
Algorithm Design and Analysis
(4)

BINUS UNIVERSITY

Study Program
Computer Science

Effective Date 01 September 2018

Revision 3

1. Course Description

The course describes fundamental concept of design and analysis of algorithms in order to calculate time and space computation, complexity, and compare design algorithm methods. It gives the students knowledge of several algorithm that enable students for designing a good algorithm.

2. Graduate Competency

Each course in the study program contributes to the graduate competencies that are divided into employability and entrepreneurial skills and study program specific outcomes, in which students need to have demonstrated by the time they complete their course.

BINUS University employability and entrepreneurial skills consist of planning and organizing, problem solving and decision making, self management, team work, communication, and initiative and enterprise.

2.1. Employability and Entrepreneurial Skills

Aspect	Key Behaviour

2.2. Study Program Specific Outcomes

Study Program Specific Outcomes

3. Topics

- Introduction of design and analysis of algorithms
- Mathematical induction and recursive function
- Algorithms and complexity
- Complexity of algorithms
- Stack and queue
- Tree and binary tree
- Priority queue and heap
- Graph
- Divide and conquer
- Greedy methods
- Dynamic Programming: Fibonacci Sequence Problem
- Dynamic Programming: Coin Change Problem
- Dynamic Programming: Multistage Graph
- Dynamic Programming: Travelling Salesman
- Dynamic Programming: Knapsack Problem
- String Matching
- Huffman Code
- Graph Colouring

- Basic Search and Traversal
- Backtracking
- Branch and Bound
- Strongly Connected Components
- Review

4. Learning Outcomes

On successful completion of this course, student will be able to:

- LO 1: Explain fundamental concept of analysis algorithms.
- LO 2: Apply algorithm techniques and methods.
- LO 3: Solve a problem using specific algorithm.
- LO 4: Compare several algorithm design methods.

5. Teaching And Learning Strategies

In this course, the lecturers might deploy several teaching learning strategis, including Lecture, Class discussion, Exercise and solve problem with students, Case Study.

6. Textbooks and Other Resources

6.1 Textbooks

1. S. Sridhar . (2015). ***Design and analysis of algorithms .*** 01. Oxford University Press . New Delhi . ISBN: 9780198093695 .
2. Thomas H. Cormen. (2009). ***Introduction to algorithms.*** 03. The MIT Press. London. ISBN: 9780262033848 .

The book in the first list is a must to have for each student.

6.2 Other Resources

1. Algorithms and complexity functions
2. Backtracking
3. Basic Search and Traversal
4. Branch and Bound
5. Complexity of algorithms analysis
6. Design and analysis of algorithms
7. Divide and conquer
8. Dynamic Programming: Coin Change Problem
9. Dynamic Programming: Fibonacci Sequence Problem
10. Dynamic Programming: Knapsack Problem
11. Dynamic Programming: Multistage Graph
12. Dynamic Programming: Travelling Salesman
13. Graph
14. Graph Colouring
15. Greedy methods
16. http://algo.is/aflv16/aflv_11_strings.pdf
17. http://algo.is/competitive-programming-course/
18. http://blogs.msdn.com/b/ericlippert/archive/2010/07/22/graph-colouring-with-simple-backtracking-part-three.
19. http://cgi.csc.liv.ac.uk/~ped/teachadmin/algor/complex.html
20. http://lmscontent.binus.ac.id/digitalcontent/DC%20COMP6226%20DAN%20COMP6049%20GRAPH%20-%
21. http://pages.cs.wisc.edu/~vernon/cs367/notes/11.PRIORITY-Q.html
22. http://people.ok.ubc.ca/ylucet/DS/KnuthMorrisPratt.html

23. http://visualgo.net/en/dfsbfs
24. http://whocouldthat.be/visualizing-string-matching/
25. http://www.algorithmist.com/index.php/Coin_Change
26. http://www.algorithmist.com/index.php/Dynamic_Programming
27. http://www.cs.cmu.edu/~adamchik/15-121/lectures/Stacks%20and%20Queues/Stacks%20and%20Queues.
28. http://www.cs.cmu.edu/~adamchik/15-121/lectures/Trees/trees.html
29. http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume4/vanbeek96a-html/node6.html
30. http://www.cs.oswego.edu/~odendahl/coursework/csc465/notes/09-c-multistage.html
31. http://www.csd.uoc.gr/~hy583/papers/ch11.pdf
32. http://www.dgp.toronto.edu/~jstewart/378notes/01intro/
33. http://www.geeksforgeeks.org/graph-and-its-representations/
34. http://www.hbmeyer.de/backtrack/achtdamen/autoacht.htm#up
35. http://www.huffmancoding.com/my-family/my-uncle/huffman-algorithm
36. http://www.ics.uci.edu/~eppstein/161/960215.html
37. http://www.imsc.res.in/~vraman/pub/intro_notes.pdf
38. http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/divide.htm
39. http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/Dynamic/knapsackdyn.htm
40. http://www.seas.gwu.edu/~ayoussef/cs6212/greedy.html
41. http://www.vogella.de/articles/ComplexityAnalysis/article.html
42. https://binus.ac.id/bits/learning-object/Complexity-of-Bubble-Sort-1217/index.html#/
43. Huffman Code
44. Introduction of design and analysis of algorithms
45. Mathematical induction and recursive function
46. Priority queue and heap
47. Review
48. Stack and queue
49. String Matching
50. Strongly Connected Components
51. Tree and binary tree

7. Schedule

Lecture

Session/Mode	Related LO	Topics	References
1 F2F	LO 1	Introduction of design and analysis of algorithms Definition of Algorithm Definition of pseudocode Introduction to analysis of algorithms	Introduction of design and analysis of algorithms Design and analysis of algorithms Some Introductionary Notes on Design and Analysis of Algorithms http://www.imsc.res.in/~vraman/pub/intro_notes.pdf

2 F2F	LO 1	Mathematical induction and recursive function Mathematics induction Recursive function	Mathematical induction and recursive function Design and analysis of algorithms Induction and Recursion http://www.imsc.res.in/~vraman/pub/intro_notes.pdf
3 F2F	LO 1	Algorithms and complexity functions Calculating processing time and growth rate Complexity function Steps top develop an algorithm	Algorithms and complexity functions Design and analysis of algorithms Computational Complexity Theory http://cgi.csc.liv.ac.uk/~ped/teachadmin/algor/complex.html
4 F2F	LO 1 LO 2	Complexity of algorithms analysis Algorithm of prime number using conventional Loop technique Algorithm of prime number using flagging Technique (sieve) Comparing both Algorithms and their complexity	Complexity of algorithms analysis Design and analysis of algorithms INTRODUCTION TO COMPLEXITY ANALYSIS http://www.dgp.toronto.edu/~jstewart/378notes/0 1intro/ Complexity Analysis of Algorithms http://www.vogella.de/articles/ComplexityAn alysis/article.html Complexity of Bubble Sort https://binus.ac.id/bits/learning-object/Complexity-of-Bubble-Sort-1217/index. html#/
5 GSLC	LO 1	Stack and queue Concept of ADT (abstract data type) Queue data structure Stack data structure	Stack and queue Design and analysis of algorithms Stack & Queue http://www.cs.cmu.edu/~adamchik/15-121/lectures/Stacks%20and%20Queues/Stacks%20and%20Queues.html

6 GSLC	LO 1	Tree and binary tree ▯ Binary tree characteristics ▯ Definition of tree ▯ Operations in tree ▯ Tree traversal	▯ Tree and binary tree ▯ Design and analysis of algorithms ▯ Binary Tree http://www.cs.cmu.edu/~adamchik/15-121/lectures/Trees/trees.html
7 F2F	LO 1 LO 2 LO 3	Priority queue and heap ▯ Concept of heap ▯ Concept of priority queue ▯ Example of priority queue in our life ▯ Heapsort ▯ Operations in heap	▯ Priority queue and heap ▯ Design and analysis of algorithms ▯ Priority Queue http://pages.cs.wisc.edu/~vernon/cs367/notes/11.PRIORITY-Q.html
8 F2F	LO 1 LO 2 LO 3	Graph ▯ Implementation of graph using cost adjacency list ▯ Implementation of graph using cost adjacency matrix ▯ Types of graph	▯ Graph ▯ Design and analysis of algorithms ▯ Graph - MST - Prim's Algorithm http://lmscontent.binus.ac.id/digitalcontent/DC%20COMP6226%20DAN%20COMP6049%20GRAPH%20-%20MST%20-%20Prim%20Algoritm.zip ▯ Graph and its representation http://www.geeksforgeeks.org/graph-and-its-representations/
9 F2F	LO1 LO2	Divide and conquer ▯ Binary search ▯ Comparing merge sort, quick sort, and selection sort ▯ Concept of divide and conquer ▯ Merge sort ▯ Quick sort ▯ Selection sort	▯ Divide and conquer ▯ Design and analysis of algorithms ▯ Divide and Conquer Algorithm http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/divide.htm
10 F2F	LO 1 LO 2	Greedy methods ▯ Disjoint Set data structure (MST) ▯ Fractional Knapsack problem ▯ Greedy method ▯ Minimum Spanning Tree ▯ Shortest Path	▯ Greedy methods ▯ Design and analysis of algorithms ▯ The Greedy Method ▯ http://www.seas.gwu.edu/~ayoussef/cs6212/gr eedy.html

11 F2F	LO 1 LO 2	Greedy methods Disjoint Set data structure (MST) Fractional Knapsack problem Greedy method Minimum Spanning Tree Shortest Path	Greedy methods Design and analysis of algorithms The Greedy Method http://www.seas.gwu.edu/~ayoussef/cs6212/greedy.html
12 F2F	LO 2 LO 3	Dynamic Programming: Fibonacci Sequence Problem Concept of Dinamic Programming Fibonacci sequence problem Solving Fibonacci Sequence problem	Dynamic Programming: Fibonacci Sequence Problem Design and analysis of algorithms Dynamic Programming http://www.algorithmist.com/index.php/Dynamic_Programming
13 F2F	LO 2 LO 3	Dynamic Programming: Coin Change Problem Coin change problem Solving of Coin Change problem	Dynamic Programming: Coin Change Problem Design and analysis of algorithms Coin Change http://www.algorithmist.com/index.php/Coin_Change
14 F2F	LO 2 LO 3	Dynamic Programming: Multistage Graph Backward technique Forward technique Multistage Graph problem	Dynamic Programming: Multistage Graph Design and analysis of algorithms Multistage Graphs http://www.cs.oswego.edu/~odendahl/coursewo rk/csc465/notes/09-c- multistage.html
15 F2F	LO 2 LO 3	Dynamic Programming: Travelling Salesman Solving of traveling salesman problem Traveling Salesman problem	Dynamic Programming: Travelling Salesman Design and analysis of algorithms The Traveling Salesman Problem http://www.csd.uoc. gr/~hy583/papers/ch11. pdf

16 F2F	LO 2 LO 3	Dynamic Programming: Knapsack Problem ▯ Comparing Greedy Method and Dynamic Programming ▯ Review of knapsack problem ▯ Solving knapsack problem using dynamic programming	▯ Dynamic Programming: Knapsack Problem ▯ Design and analysis of algorithms ▯ Dynamic-Programming Solution to the 0-1 Knapsack Problem http://www.personal.kent. edu/~rmuhamma/Algorithms/MyAlgorithms/Dynami ▯ c/knapsackdyn.htm
17 GSLC	LO 2 LO 3	String Matching ▯ Naïve String Matching Algorithm ▯ The Knuth-Morris-Pratt Algorithm	▯ String Matching ▯ Design and analysis of algorithms ▯ Knuth-Morris-Pratt String Search http://people.ok.ubc. ca/ylucet/DS/KnuthMorris Pratt.html ▯ String Algorithm http://whocouldthat. be/visualizing-string-matching/ ▯ String http://algo. is/aflv16/aflv_11_strings.pdf
18 GSLC	LO 2 LO 3	Huffman Code ▯ Building Huffman tree ▯ Concept of compression technique ▯ Creating Huffman Code table based on Huffman tree ▯ Introduction to Huffman code	▯ Huffman Code ▯ Design and analysis of algorithms ▯ Huffman Coding http://www.huffmancoding.com/my-family/myuncle/huffman-algorithm
19 F2F	LO 2 LO 3	Graph Colouring ▯ Chromatic number ▯ Edge Colouring ▯ Graph Colouring ▯ Map colouring ▯ Node Colouring ▯ Region Colouring	▯ Graph Colouring ▯ Design and analysis of algorithms ▯ Graph Colouring with Simple Backtracking, Part Three http://blogs.msdn. com/b/ericlippert/archive/ 2010/07/22/graph-colouring-with-simple-backtracking-part-three. aspx

20 F2F	LO 2 LO 3 LO 4	Basic Search and Traversal BFS in Tree and Graph Breadth First Search Depth First Search DFS in Tree and Graph Tree traversal	Basic Search and Traversal Design and analysis of algorithms BFS dan DFS http://www.ics.uci.edu/~eppstein/161/960215.html
21 GSLC	LO 2 LO 3	Backtracking Concept of Backtracking N-Queen Problem	Backtracking Design and analysis of algorithms Backtracking Algorithm http://www.hbmeyer.de/backtrack/achtdamen/ autoacht.htm#up Backtracking Algorithm http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume4/vanbeek96a-html/node6.html
22 GSLC	LO 2 LO 3	Branch and Bound Definition of branch and bound FIFO branch and bound LC branch and bound LIFO branch and bound Solving traveling salesman problem	Branch and Bound Design and analysis of algorithms
23 F2F	LO 2 LO 3	Branch and Bound Definition of branch and bound FIFO branch and bound LC branch and bound LIFO branch and bound Solving traveling salesman problem	Branch and Bound Design and analysis of algorithms
24 F2F	LO 2 LO 3	Strongly Connected Components Strongly connected definition Tarjan's strongly connected component algorithm	Strongly Connected Components Design and analysis of algorithms Strongly Connected Components Algorithm http://visualgo.net/en/dfsbfs

25 F2F	LO 2 LO 3 LO 4	Review Dynamic Programming String Matching Strongly Connected Component	Review Design and analysis of algorithms Practices of algortihm analysis http://algo.is/competitive- programming-course/
26 F2F	LO 2 LO 3 LO 4	Review Dynamic Programming String Matching Strongly Connected Component	Review Design and analysis of algorithms Practices of algortihm analysis http://algo.is/competitive-programming-course/

8. Evaluation

Lecture

Assessment Activity	LO			
	1	2	3	4
ASSIGNMENT	✓	✓	✓	✓
FINAL EXAM		✓	✓	✓
MID EXAM	✓	✓	✓	

Final Evaluation Score

Aspects	Weight
Theory	100%

9. Assessment Rubric (Study Program Specific Outcomes)

LO	Indicators	Proficiency Level			
		Excellent (85 - 100)	**Good (75 - 84)**	**Average (65 - 74)**	**Poor (<= 64)**
LO 1	1.1. Ability to define the concept of analysis algorithm	The concept of analysis algorithm is clearly defined and relevant	The concept of analysis algorithm is clearly defined	The concept of analysis algorithm is not clearly defined	The concept of analysis algorithm is defined incorrectly
	1.2. Ability to demonstrate the stack and Queue applications and operations	Correctly and effectively demonstrating the stack and queue applications and operations	Correctly demonstrating the stack and queue applications and operations	Partially correct in demonstrating the stack and queue applications and operations	Incorrectly demonstrating the stack and queue applications and operations

LO 2	2.1. Ability to explain the algorithm techniques and methods	Correct explanation is given with relevant examples	Correct explanation is given with partially relevant examples	Correct explanation	Incorrect explanation
	2.2. Ability to give examples how to use the algorithm techniques and methods	All the examples are align to the algorithm techniques and methods	Only some of the examples are align to the algorithm techniques and methods	Only few of the examples are align to the algorithm techniques and methods	none of the examples are align to the algorithm techniques and methods
LO 3	3.1. Ability to calculate the complexity of an algorithm	Correctly and effectively calculating the complexity of an algorithm	Correctly calculating the complexity of an algorithm	Partially correct in calculating the complexity of an algorithm	Incorrectly calculating the complexity of an algorithm
	3.2. Ability to design algorithm with specific complexity	Correctly and effectively designing algorithm with specific complexity	Correctly designing algorithm with specific complexity	Partially correct in designing algorithm with specific complexity	Incorrectly designing algorithm with specific complexity
LO 4	4.1. Ability to interpret the algorithm design methods	Correctly interpreting all of the algorithms design methods	Correctly interpreting many of the algorithms design methods	Correctly interpreting some of the algorithms design methods	Incorrectly interpreting the algorithms design methods
	4.2. Ability to choose the algorithm design method for a real case	Correctly choosing the most effective algorithm design method for a real case	Correctly choosing effective algorithm design method for a real case	Correctly choosing algorithm design method for a real case	Incorrectly choosing algorithm design method for a real case

Prepared by **D5542 - Fidelson Tanzil, S.Kom., M.TI**	Checked by **D5542 - Fidelson Tanzil, S.Kom., M.TI** **Acting as Subject Content Specialist**
Approved by **D5542 - Fidelson Tanzil, S.Kom., M.TI** **Acting as Subject Content Coordinator**	Acknowledged by **D3690 - Derwin Suhartono, S.Kom., M.T.I.** **Head of Program – Computer Science**