Akashi College	Year	2022	Course Title	Algorithms

Course Information

Course Code	4036	Course Category	Specialized / Elective
Class Format	Lecture	Credits	Academic Credit: 2
Department	Mechanical and Electronic System Engineering	Student Grade	Adv. 2nd
Term	Second Semester	Classes per Week	2
Textbook and/or Teaching Materials			
Instructor	HAMADA Yukihiro		

Course Objectives

[1] Can explain the basic knowledge of algorithms and the basic data structure (D).
[2] Can formulate real problems on graphs (F).
Understand the algorithms listed below and their time complexities (H).
[3] Algorithms that constitute a minimum spanning tree
[4] Algorithms to explore graphs
[5] Algorithms for solving shortest path problem
[6] Algorithms for solving maximum flow problems
[7] Algorithms for string pattern matching

Rubric

	Ideal Level	Standard Level	Unacceptable Level
Achievement 1	Can accurately explain computational complexity, orders, lists, stacks, queues, graphs, and trees.	Can explain computational complexity, orders, lists, stacks, queues, graphs, and trees.	Cannot explain computational complexity, orders, lists, stacks, queues, graphs, and trees.
Achievement 2	Can accurately formulate a problem for determining the meeting dates of various committees.	Can formulate a problem for determining the meeting dates of various committees.	Cannot formulate a problem for determining the meeting dates of various committees.
Achievement 3	Can accurately explain Kruskal's and Prim's algorithms and their time complexities.	Can explain Kruskal's and Prim's algorithms and their time complexities.	Cannot explain Kruskal's and Prim's algorithms and their time complexities.
	Can accurately explain depth-first search and breadth-first search algorithms and their time complexities.	Can explain depth-first search and breadth-first search algorithms and their time complexities.	Cannot explain depth-first search and breadth-first search algorithms and their time complexities.
	Can accurately explain Dijkstra's, Bellman-Ford, and Floyd's algorithms and their time complexities.	Can explain Dijkstra's, Bellman-Ford, and Floyd's algorithms and their time complexities.	Cannot explain Dijkstra's, Bellman-Ford, and Floyd's algorithms and their time complexities.
	Can accurately explain the Ford-Fulkerson, Edmonds-Karp, and Push-relabel algorithms and their time complexities.	Can explain the Ford-Fulkerson, Edmonds-Karp, and Push-relabel algorithms and their time complexities.	Cannot explain the Ford-Fulkerson, Edmonds-Karp, and Push-relabel algorithms and their time complexities.
	Can accurately explain the Knuth-Morris-Pratt and Boyer-Moore algorithms and their time complexities.	Can explain the Knuth-Morris-Pratt and Boyer-Moore algorithms and their time complexities.	Cannot explain the Knuth-Morris-Pratt and Boyer-Moore algorithms and their time complexities.

Assigned Department Objectives

Teaching Method

Outline	This course will study graph algorithms and string pattern matching algorithms. Graphs are defined as binomial sets of vertex and edge sets, and are often used to represent the "relationships" or "connections" between "things" in real-world problems. It is possible to formulate a real problem as a graph problem and get the solution for it by solving it on a graph. Strings are one of the most important kinds of data handled by computers. Students will learn about algorithms for efficiently finding specified strings in string data, such as documents or source files.
Style	Classes will be held in a lecture style.
Notice	This course's content will amount to 90 hours of study in total. These hours include the learning time guaranteed in classes and the standard self-study time required for pre-study / review, and completing assignment reports. It is recommended for students to have mastered programming in C language before taking this course. Students who miss 1/3 or more of classes will not be eligible for a passing grade.

Characteristics of Class / Division in Learning

☐ Active Learning	☑ Aided by ICT	☑ Applicable to Remote Class	☐ Instructor Professionally Experienced

Course Plan

			Theme	Goals
2nd Semester	3rd Quarter	1st	Basic knowledge of algorithms	Can explain algorithms, computational complexity, and orders.
		2nd	Basic data structure	Can explain lists, stacks, queues, and heaps.
		3rd	How to formulate real-world problems as graph problems	Can explain graphs and trees. Can formulate a problem for determining the meeting dates of various committees as a problem on a graph.
		4th	Algorithms that constitute a minimum spanning tree algorithm 1/2	Can explain Kruskal's algorithm, set operation algorithms and their time complexities.

		5th	Algorithms that constitute a minimum spanning tree 2/2	Can explain Prim's algorithm and its time complexity.
		6th	Algorithms to explore graphs	Can explain depth-first search and breadth-first search algorithms and their time complexities.
		7th	Algorithms for solving shortest path problems 1/2	Can explain Dijkstra's algorithm for finding the shortest path from a single vertex and its time complexity.
		8th	Midterm exam The exam's scope will be content from weeks 1 to 6.	
	4th Quarter	9th	Algorithms for solving shortest path problems 2/2	Can explain the Bellman-Ford algorithm for the shortest path from a single vertex and the Floyd algorithm for the shortest path between all vertices. Can also explain their time complexities.
		10th	Algorithms for solving maximum flow problems 1/2	Can explain the Ford-Fulkerson and Edmonds-Karp algorithms and their time complexities.
		11th	Algorithms for solving maximum flow problems 2/2	Can explain the Push-relabel algorithm and its time complexity.
		12th	Algorithms for string pattern matching 1/3	Can explain the Knuth-Morris-Pratt algorithm and its time complexity.
		13th	Algorithms for string pattern matching 2/3	Can explain the Boyer-Moore algorithm (acceleration idea 1) and its time complexity.
		14th	Algorithms for string pattern matching 3/3	Can explain the Boyer-Moore algorithm (acceleration idea 2) and its time complexity.
		15th	From algorithm theory to engineering	Can explain "algorithm engineering," which bridges the gap between algorithm theory and reality.
		16th	Final exam	

Evaluation Method and Weight (%)

	Examination	Presentation	Mutual Evaluations between students	Behavior	Portfolio	Other	Total
Subtotal	100	0	0	0	0	0	100
Basic Proficiency	0	0	0	0	0	0	0
Specialized Proficiency	100	0	0	0	0	0	100
Cross Area Proficiency	0	0	0	0	0	0	0