

Course Description					
Name	Code	Semester	T+A Hour	Credit	ECTS
ALGORITHM ANALYSIS	BME3233890	Spring Semester	3+0	3	6
Prerequisites Courses	OLASILIK VE RASSAL DEĞİŞKENLER; VERİ YAPILARI				
Recommended Elective Courses	Data Structures				
Language of Instruction	English				
Course Level	First Cycle (Bachelor's Degree)				
Course Type	Elective				
Course Coordinator	Prof.Dr. Reda ALHAJJ				
Name of Lecturer(s)	Prof.Dr. Reda ALHAJJ				
Assistant(s)					
Aim	Introduce fundamental techniques for designing algorithms and analyzing the time and space requirements of these algorithms in a formal way. Mathematical background for algorithm analysis, sorting, searching, basic algorithms design and graph algorithms will be covered.				
Course Content	This course contains; Week 1: Introduction: analysing algorithms, designing algorithms.,Week 2: Asymptotic Notation.,Week 3: Divide and Conquer Design Paradigm.,Week 4: Solving Recurrences.,Week 5: Analysis of Quicksort, Randomized Quicksort.,Week 6: Heapsort.,Week 7: Quicksort.,Week 8: Sorting in Linear Time.,Midterm,Week 10: Medians and Order Statistics.,Week 11: Dynamic Programming.,Week 12: Greedy Algorithms.,Week 13: Amortized Analysis, Dynamic Tables.,Week 13: Graphs, Breadth-first Search (BFS)..				
Course Learning Outcomes			Teaching Methods	Assessment Methods	
1) At the end of this course the students will be able to describe the fundamentals of algorithm analysis.			12, 14, 16, 9	A, E	
2) At the end of this course the students will be able to construct complex algorithms using the data structures that they have learned.			12, 14, 16, 9	A, E	
3) At the end of this course the students will be able to develop complex algorithms and advanced data structures that are using trees and will be able to apply them to real world problems.			10, 12, 14, 17, 9	A, E, F	
4) At the end of this course the students will be able to develop complex algorithms and advanced data structures that are using graphs and will be able to apply them to real world problems.			10, 12, 14, 17, 9	A, E, F	
5) At the end of this course the students will be able to systematically look at a given computational problem and design a novel algorithm using techniques like dynamic programming, divide and conquer and greedy algorithms.			12, 14, 16, 19, 9	A, E	
Teaching Methods	10: Discussion Method, 12: Problem Solving Method, 14: Self Study Method, 16: Question - Answer Technique, 17: Experimental Technique, 19: Brainstorming Technique, 9: Lecture Method				
Assessment Methods	A: Traditional Written Exam, E: Homework, F: Project Task				
Lecture Schedule					
Sequence	Topics	Preliminary Preparation			
1	Week 1: Introduction: analysing algorithms, designing algorithms.	Lecture Slides and textbook chapters 1 & 2			
2	Week 2: Asymptotic Notation.	Lecture Slides and textbook chapter 3			
3	Week 3: Divide and Conquer Design Paradigm.	Lecture Slides and textbook chapter 4			
4	Week 4: Solving Recurrences.	Lecture Slides and textbook chapter 4			
5	Week 5: Analysis of Quicksort, Randomized Quicksort.	Lecture Slides and textbook chapter 5			
6	Week 6: Heapsort.	Lecture Slides and textbook chapter 6			
7	Week 7: Quicksort.	Lecture Slides and textbook chapter 7			
8	Week 8: Sorting in Linear Time.	Lecture Slides and textbook chapter 8			
9	Midterm	Lecture Slides and textbook chapters from 1 to 9, inclusive.			
10	Week 10: Medians and Order Statistics.	Lecture Slides and textbook chapter 9			
11	Week 11: Dynamic Programming.	Lecture Slides and textbook chapter 15			
12	Week 12: Greedy Algorithms.	Lecture Slides and textbook chapter 16			
13	Week 13: Amortized Analysis, Dynamic Tables.	Lecture Slides and textbook chapter 17			
14	Week 13: Graphs, Breadth-first Search (BFS).	Lecture Slides and textbook chapter 22			
Evaluation Methods		Weight(%)			
Midterm Exam		30			
General Exam		70			

Resources
T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, Mit Press and McGraw-Hill, 2009. The notes and the presentations will be delivered during the lectures.