

Algorithm

Course Name	Course section (credit/hours)		Required course(3/3)			course code	F046
	course item					course component	
	Target students Division/major/grade					opening semester	2021 1ST SEMESTER
	Class time and classroom		Mon D(Pal309)Thu D(Pal309)			English Grade	A(100%English)
Reference to this course	Credit compositon		Theory(3) + Design(0) + Practice(0)				
	Prerequisite courses		자료구조 (Data Structure)				
	Related basic courses		이산수학 (Discrete Math)				
	Recomanded concurrent courses		인공지능 (Artificial Intelligence)				
	Related advanced course		계산이론 (Theory of Computation)				
Instructor	Name (title/division)		Sael Lee(Associate Professor, AI and Data Science)				
	Office Room Number	산학원 620호	Extension Number	0312193839	e-mail	sael@ajou.ac.kr	
	Office hour	월(Mon) 3:00-5:00		Homepage address	dilab.ajou.ac.kr		
Teaching Assistant	Name (title/division)						
	Office Room Number		Office phone Number		e-mail		

1. Course Introduction

This course deals with principles and techniques for design and analysis of computer algorithms. The topics covered are mathematical induction, asymptotic analysis of algorithm efficiency, and algorithm design techniques including divide-and-conquer, dynamic programming, greedy method, branch-and-bound, backtracking, and iterative improvements. Elements of computational complexity theory, mostly on NP-completeness, is introduced and it is also discussed how to cope with computationally intractable problems.

2. Course Objectives & course outcome

어떠한 응용분야이든지 좋은 컴퓨터 소프트웨어를 작성하거나 개발하려면 효율적인 알고리즘에 기반해야 한다. 컴퓨터 하드웨어가 아무리 우수하더라도 소프트웨어가 비효율적인 알고리즘에 기반하고 있으면 효율적인 정보처리를 기대할 수 없다. 본 과목에서는 주어진 알고리즘의 효율성을 분석하는 원리 및 기술을 학습하고, 또한 효율적인 알고리즘을 설계하는 기법들을 배운다. 졸업 후에 취업을 하든지 대학원에 진학을 하든지 프로그램을 작성하는 능력은 누구나 기본적으로 갖추고 있어야 한다. 어떠한 문제를 해결하는 프로그램을 작성하게 될지는 미리 알 수 없다. 업무를 수행하거나 연구를 수행할 때, 이미 잘 알려진 문제를 해결해야 하는 경우라면 잘 알려진 알고리즘이 있을 것이며, 본 과목에서 배운 알고리즘을 사용할 수 있을 것이다. 새로운 문제를 해결해야 하는 경우라면 본 과목에서 학습한 효율적인 알고리즘의 설계 기법을 적용할 수 있을 것이다.

The goal of this course is to enable students to recognize, analyze, and solve algorithmic problems. At the end of the course, students should be able apply core algorithmic problems that underlie many programming tasks, identify and use appropriate algorithmic techniques to solve those problems, and analyze and compare the performance of algorithmic solutions.

본 과목을 수강하는 3, 4 학년생들은 전산학의 다양한 분야의 과목들을 이미 수강하여 그 분야에서 다루는 문제들과 그 해법들에 익숙할 것이다. 또한 여러 분야에서 공통으로 다루는 문제가 있다는 사실을 알 것이며, 또한 겉으로는 달라 보이는 문제들이지만 본질적으로는 같은 문제임을 인지하는 경우도 있을 것이다. 본 과목에서는 이러한 문제들을 추상화하고 이들을 해결하는 알고리즘 또는 알고리즘을 설계하는 기법을 학습함으로써 전산학의 다양한 분야의 연관관계를 이해하고 이들을 관통하는 조망을 얻는 기회가 될 것이다.

컴퓨터 알고리즘의 디자인과 분석을 위한 원리와 기법을 학습하여 실제의 문제들을 해결하는 효율적인 알고리즘들을 설계하고 구현할 수 있는 능력을 기를 수 있도록 한다. 또한 알고리즘의 성능을 분석하고, 알고리즘의 성능을 비교하고, 알고리즘의 성능을 평가할 수 있는 능력을 기를 수 있도록 한다.

3. Class types and activities

Mostly lectures.

Assignments consist of exercise problems on algorithm efficiency analysis, algorithm designs, and algorithm correctness. Students are supposed to invest considerable amount of time to understand course material and to solve assignment problems.

4. Teaching Method

- | | |
|--|---|
| <input checked="" type="checkbox"/> lecture | <input type="checkbox"/> discussion and debate |
| <input type="checkbox"/> team project(presentation and case studies) | <input type="checkbox"/> experiments(role-playing, etc) |
| <input type="checkbox"/> designing and production | <input type="checkbox"/> on-site learning(on-site training) |
| <input type="checkbox"/> others | |

5. Support Systems in Use

- | | | |
|--|---|---|
| <input checked="" type="checkbox"/> AjouBb | <input type="checkbox"/> automatic recording system | <input type="checkbox"/> web-based assignment |
| <input type="checkbox"/> cyber lecture | <input type="checkbox"/> online content | |
| <input type="checkbox"/> class behavior analyzing system | <input type="checkbox"/> others | |

6. Teaching Tools

- | | | |
|--|---|---|
| <input type="checkbox"/> PBL(Problem Based Learning) | <input type="checkbox"/> CBL(Case Based Learning) | <input type="checkbox"/> TBL(Team Based Learning) |
| <input type="checkbox"/> UR(Undergraduate Research) | <input type="checkbox"/> FL(Flipped Learning) | <input type="checkbox"/> DSAL(Data Science Active Learning) |
| <input type="checkbox"/> others | | |

7. Evaluation method of course outcome

Evaluation Item	The Number of Times	Evaluation Proportion	Remarks
Attendance		5%	
midterm exam	2	50%	
final exam	1	30%	전범위/accumulative – all topics learned during the semester
quiz			

7. Evaluation method of course outcome

Evaluation Item	The Number of Times	Evaluation Proportion	Remarks
presentation			
discussion			
homework	5-6	15	done with 4-5 group members.
etc			
study hours	5		

8. Textbook and Reference material

Main/Sub	Title	Writer	Publisher	Publication year
Main	Foundations of Algorithms, 5th edition	Richard Neapolitan	Jones & Bartlett	2015
Sub	The Algorithm Design Manual	Steven S. Skiena	Springer	2008

9. Class system and Class shedule

In the beginning of the course, concepts of algorithms, mathematical induction, asymptotic analysis are taught. The algorithm design techniques follow including divide-and-conquer, dynamic programming, greedy method, and iterative improvements. Then the students will learn that there are problems that do not have efficient algorithms, and how to cope with such problems.

강의 초반에는 알고리즘의 정의, 수학적 귀납법, 알고리즘 효율성의 점근적 분석법 등을 배운다. 그 다음에는 분할정복, 동적 계획법, 그리디 방법, 퇴각검색, 분지한정 등의 알고리즘 설계 기법을 공부한다. 강의 후반에는 효율적인 알고리즘이 존재하지 않는 문제들이 있다는 사실을 배우고, 그러한 문제들을 다루는 방법에 대해서 공부한다.

< Schedule >

* language : K-korean, E-English

Weeks	Title of lecture	language	time distribution(minutes)			Teaching Method	evaluation method
			theory	design	experiment practice		
1	Introduction	E					
2	Analysis of algorithm efficiency	E					
3	Divide-and-Conquer	E					
4	Divide-and-Conquer	E					
5	Sorting and Searching	E					
6	Midterm I / Midterm I Overview	E					

< Schedule >

* language : K-korean, E-English

Weeks	Title of lecture	language	time distribution(minutes)			Teaching Method	evaluation method
			theory	design	experiment practice		
7	Dynamic Programming	E					
8	Dynamic Programming	E					
9	Greedy Approaches	E					
10	Geedy Approaches	E					
11	Midterm Exam II / Midterm II Overview	E					
12	Graph Algorithms	E					
13	Backtracking	E					
14	Branch-and-Bound	E					
15	Intractable Problems/NP-hard Problems	E					
16	Final Exam	E					

10. Contribution index of the course for attaining ABEEK program outcomes

course outcome	contribution scale
No Data	

11. Analysis of improved matters for the previous semester

13. Reference items

중간고사 I & II, 기말고사는 비대면으로 진행한다. 중간고사 I & II 는 수업시간에 진행된다.