

Syllabus

Advanced Algorithm

Prof.	Name	Kyubum Wee	Sub.	Student	Department	Computer Engineering
	Position	Professor			Major	Computer Engineering
	Group	Software and Computer Engineering				

1. Course Description

Design and analysis of efficient computer algorithms are studied. The topics covered are graph algorithms, geometric algorithms, string matching algorithms, randomized algorithms, NP-completeness and approximation algorithms, and genetic algorithms.

Before enrolling in this course, you should carefully read 기타 참고사항 (Others) part of this syllabus.

2. Teaching Methods

Mostly Lectures

3. Evaluation

4. TextBooks

Text : "Introduction to Algorithms" (second edition), Cormen, Leiserson, Rivest, and Stein, MIT Press, 2001.

References: (1) "Algorithms", Johnsonbaugh and Schaeffer, Prentice-Hall, 2004.

(2) "Fundamentals of Algorithmics", Brassard and Bratley, Prentice-Hall, 1996.

5. Lecture Schedule

Week	Lecture contents	Lesson type	Remark
1	Strongly connected components	Lecture	Dasgupta chapter 3 (Cormen chapter 22)
2	Biconnected components	Lecture	Dasgupta chapter 3 (Cormen chapter 22)
3	Flow networks, Ford–Fulkerson method	Lecture	Cormen chapter 26 (Dasgupta chapter 7)
4	Max Flow Min Cut Theorem, Bipartite matching	Lecture	Cormen chapter 26 (Dasgupta chapter 7)
5	Line–segment properties	Lecture	Cormen chapter 33
6	Convex hulls	Lecture	Cormen chapter 33
7	Closest pair of points	Lecture	Cormen chapter 33
8	Midterm Exam		
9	Rabin–Karp string matching	Lecture	Cormen chapter 32
10	Knuth–Morris–Pratt string matching	Lecture	Cormen chapter 32
11	Number–theoretic concepts	Lecture	Cormen chapter 31
12	Randomized primality testing	Lecture	Cormen chapter 31
13	NP–completeness	Lecture	Neapolitan chapter 9 (Cormen chapter 34)
14	NP–completeness proofs	Lecture	Neapolitan chapter 9 (Cormen chapter 34)
15	Approximation algorithms	Lecture	Neapolitan chapter 9 (Cormen chapter 35)
16	Final Exam		

6. Others

This is a graduate–level course. If you have not taken an undergraduate–level algorithms course, you are not allowed to take this course.

Students are required to have taken the undergraduate–level algorithms course "SCE332 Algorithms" or a course which is equivalent to SCE332. A student is required to present the relevant materials that show the course is equivalent to SCE332.

I will not accept a student who has not taken an undergraduate–level algorithms course. If you have not taken SCE332 or equivalent course, you should take SCE332 instead, not this one.

If you have not taken SCE332 or equivalent one, your taking this course, which I will not allow, would be total waste of your time and effort as well as mine.