San José State University
Computer Science Department
CS255, Section 1, Design and Analysis of Algorithms, Spring 2020

Course and Contact Information

Instructor(s):	Aikaterini Potika
Office Location:	MacQuarrie Hall 215
Telephone:	408-9245134
Email:	katerina.potika@sjsu.edu
Office Hours:	T 1:30-3pm and Th 9:45-10:15am am or by appointment
Class Days/Time:	TTh 12-1:15 pm
Classroom:	MacQuarrie Hall 222
Prerequisites:	CS 155 or instructor consent

Course Description

Randomized algorithms. Parallel algorithms. Distributed algorithms. NP-completeness of particular problems. Approximation algorithms.

Course Format

Faculty Web Page and MYSJSU Messaging

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on Canvas Learning Management System course login website at http://sjsu.instructure.com. You are responsible for regularly checking with the messaging system through MySJSU on Spartan App Portal http://one.sjsu.edu (or other communication system as indicated by the instructor) to learn of any updates. For help with using Canvas see Canvas Student Resources page (http://www.sjsu.edu/ecampus/teaching-tools/canvas/student_resources).

Course Learning Outcomes (CLO)

Upon successful completion of this course, students will be able to:

1. CLO 1. Code an example of each of the following types of algorithms:

a. Randomized

b. Parallel

c. Approximation

2. CLO 2. Conduct an amortized analysis.

3. CLO 3. Explain how above techniques are used in several applications, and describe what benefits they have within those applications.

Required Texts/Readings

Textbook

No required textbook we will use chapters from various books:

1. Cormen, Leiserson, Rivest and Stein, Introduction to Algorithms, 3rd Edition
   MIT Press, 2009. You can find errata (bug reports) for the book http://www.cs.dartmouth.edu/~thc/clrs-bugs/bugs-3e.php.
2. Kleinberg and Tardos, Algorithm Design, First edition, Addison Wesley, 2005.
3. Dasgupta, Papadimitriou and Vazirani, Algorithms, McGraw-Hill, 2006.
4. Vazirani, Approximation Algorithms, Springer, 2003

Other Readings

- Research papers
- Handouts (through Canvas)

Course Requirements and Assignments

Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus.

Homework assignments: individual, regularly assigned, will include written problem assignments, and perhaps some online exercises. Solutions will be not posted. The homework is a tool for you to learn the material and prepare you for the exams.

Reading assignments: Reading assignments are regular and for the next class.

Quizzes: Unannounced quizzes (at least 4) may be given during class, each taking about 5 minutes total. These generally are problems from the reading assignment and/or the homework.

Project (Programming and Presentation): A programming project of your choice related to the course in groups of two students. At the end of the semester you will present the project in the class. Never use any code you find on the web, unless it is given by me. Penalty for late submission 5% for every 3 days up to 9 days, after that no submission will be accepted. Never email your assignments.

Midterm exams: Two written Midterm exams during the semester.

Final Examination or Evaluation

One final, written, and cumulative exam, split in two parts. The exams contain multiple-choice questions, short answer questions and questions that require pseudocode and/or computations.

Grading Information

No extra point options (only the final exam offers extra points option).
All exams are closed book, and final exam is comprehensive. No make-ups exams except in case of verifiable emergency circumstances.

Determination of Grades

Final Grade:
25% Project (programming and presentation)
5% Quizzes
10% Homework
30% Midterms (15% each)
30% Final

A+	A	A-	>90
B+	B	B-	>78
C+	C	C-	>65
D+	D	D-	>45
F			<40

Classroom Protocol

Attendance is highly recommended. Please avoid disturbing the class: turn-off cell phones (or put them on vibrate mode), no text messaging in the class or the exams, no taking pictures and video, avoid coming late, no talking or whispering with other students during instructor's presentation. You may not publicly share or upload material of this course such as exam questions, lecture notes, or solutions without my consent.

University Policies

Per University Policy S16-9 *(http://www.sjsu.edu/senate/docs/S16-9.pdf)*, relevant university policy concerning all courses, such as student responsibilities, academic integrity, accommodations, dropping and adding, consent for recording of class, etc. and available student services (e.g. learning assistance, counseling, and other resources) are listed on Syllabus Information web page (http://www.sjsu.edu/gup/syllabusinfo), which is hosted by the Office of Undergraduate Education. Make sure to visit this page to review and be aware of these university policies and resources.

The instructor might drop students that do not show up during the first two lectures.

CS255: Design and Analysis of Algorithms, Spring 2020

The schedule is subject to change with fair notice and announced on Canvas.

Course Schedule

Lesson	Date	Topic	Assignments
1	1/23	Introduction: Algorithms & Computers	
2	1/28	Some Examples, Stable Matching	HW 1
3	1/30	Running time, growth of functions	
4	2/4(drop)	Graphs, BFS, DFS, topological sorting	
5	2/6	Number-Theoretic Algorithms, Searching	
6	2/11 (add)	Greedy Algorithms: Scheduling, Shortest paths, Caching, knapsack	
7	2/13	Greedy Algorithms: Minimum spanning tree, clustering	
8	2/18	Divide & Conquer: sorting, integer/matrix multiplication, max subarray	
9	2/20	Divide & Conquer: computational geometry	
10	2/25	Dynamic Programming: scheduling, knapsack	
11	2/27	Dynamic Programming: all pair shortest path	
12	3/3	Network flow, applications	
	3/5	**Midterm 1**	
13	3/10	Heaps, Amortized Analysis	Project proposal (due)
14	3/12	Amortized Analysis cont.	
15	3/17	Randomization: Quicksort	
16	3/19	Randomization: Hashing	

17	3/24	Intractability, P, NP, NP-completeness,	
18	3/26	Intractability, P, NP, NP-completeness, reductions, time hierarchy	
	3/31-4/2	*Spring break*	
19	4/7	Intractability, P, NP, NP-completeness, reductions, time hierarchy	Project demo (due)
20	4/9	Intractability, P, NP, NP-completeness, reductions, time hierarchy	
21	4/14	Linear Programming, Simplex	
	4/16	**Midterm 2**	
22	4/21	Approximation Algorithms: vertex cover, TSP	
23	4/23	Approximation Algorithms: path coloring	Project final (due)
24	4/28	Distributed Algorithms: Byzantine Agreement	
25	4/30	Project Presentations	
26	5/5	Project Presentations	
27	5/7	Project Presentations	
		Final exam Wednesday, May 13 | 09:45-12:00	