San José State University
Computer Science Department

CS146, Sections 6 and 8, Data Structures and Algorithms, Fall 2019

Course and Contact Information

Instructor:	William “Bill” Andreopoulos
Office Location:	MacQuarrie Hall 416
Telephone:	408-924-5085
Email:	william.andreopoulos@sjsu.edu
Office Hours:	Monday 12:00-13:00 pm and Wednesday 15:00-16:15
Class Days/Time:	MW 10:30-11:45 am (Section 6) and 16:30-:17:45 pm (Section 8)
Classroom:	MacQuarrie Hall 222 (Section 6) and MacQuarrie Hall 223 (Section 8)
Prerequisites:	MATH 030, MATH 042, CS 049J (or equivalent knowledge of Java), and/or CS 046B (with a grade of "C-" or better in each); or instructor’s consent.

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 at http://my.sjsu.edu (or other communication system as indicated by the instructor) to learn of any updates.

Course Description

Implementations of advanced tree structures, priority queues, heaps, directed and undirected graphs. Advanced searching and sorting (radix sort, heapsort, mergesort, and quicksort). Design and analysis of data structures and algorithms. Divide-and-conquer, greedy, and dynamic programming algorithm design techniques.

Course Learning Outcomes (CLO)

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

1. CLO1. Understand the implementation of lists, stacks, queues, search trees, heaps, union-find ADT, and graphs and be able to use these data structures in programs they design
2. CLO2. Prove basic properties of trees and graphs

3. CLO3. Perform breadth-first search and depth-first search on directed as well as undirected graphs
4. CLO4. Use advanced sorting techniques (heapsort, mergesort, quicksort)
4. CLO5. Determine the running time of an algorithm in terms of asymptotic notation
5. CLO6. Solve recurrence relations representing the running time of an algorithm designed using a divide-and-conquer strategy
6. CLO7. Understand the basic concept of NP-completeness and realize that they may not be able to efficiently solve all problems they encounter in their careers
7. CLO8. Understand algorithms designed using greedy, divide-and-conquer, and dynamic programming techniques

Required Texts/Readings

Textbooks

Cormen, Leiserson, Rivest and Stein, Introduction to Algorithms, 3rd
Edition. (*CLRS*)
ISBN-13: 978-0262033848
ISBN-10: 9780262033848
MIT Press, 2009
You can find errata (bug reports) for the book:
http://www.cs.dartmouth.edu/~thc/clrs-bugs/bugs-3e.php.
CLRS e-textbook is available via the SJSU library:
https://sjsu-primo.hosted.exlibrisgroup.com/permalink/f/1cue0e3/01CALS_ALMA51438951350002901

Sedgewick and Wayne, Algorithms, 4th Edition. (*SW*)
ISBN-13: 978-0321573513
ISBN-10: 032157351X
This book and its website contain Java implementation of many algorithms covered:
https://algs4.cs.princeton.edu/code/

These books will be on reserve in the library

Other Readings

- Horstmann and Cornell, Core Java, Vol. I, Ninth edition, Prentice Hall, 2013.
- Kleinberg and Tardos, Algorithm Design, First edition, Addison Wesley, 2005.
- Dasgupta, Papadimitriou and Vazirani, Algorithms, McGraw-Hill, 2006.
- Handouts (through Canvas)

Other technology requirements / equipment / material

Java Compiler (version 7 or later) and Eclipse.

Course Requirements and Assignments

SJSU classes are designed such that in order to be successful, it is expected that students will spend a minimum of forty-five hours for each unit of credit (normally three hours per unit per week), including preparing for class, participating in course activities, completing assignments, and so on.

Homework pre-class assignments: individual, regularly assigned, not graded, will include written problem assignments, and perhaps some online exercises. We will go through sample solutions in class. The homework is a tool for you to learn the material and prepare for the exams.

In-class problems: there will be in-class problem solving and quizzes, not graded, to be done with the assistance of the embedded tutor. These will be done on Canvas and it is recommended to bring a laptop. These will generally be problems from the reading assignment and/or the homework. These problems are a tool for you to learn the material and prepare for the exams.

Reading assignments: Readings will regularly be assigned for the next class (see schedule).

Programming assignments: Programming assignments will be assigned. Programming assignments are done individually, unless otherwise specified. Students can discuss them, but they should be implemented individually. More information will be given at the time of the first programming assignment. Never copy any code you find on the web. Penalty for late submission 5% for every 3 days up to 15 days, after that no submission will be accepted. Never email your assignments, always upload to Canvas. Oral examination might be requested.

Midterm exams: There will be two written Midterm exams during the semester.

Final Examination or Evaluation

Final exam: One written final cumulative exam.

The exams will contain multiple choice questions, short answer questions and questions that require pseudocode and/or computations.

Exams are *closed book*, final exam is comprehensive. No make-up exams except in case of verifiable emergency circumstances.

Piazza for Class Discussion

This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. Find our class Piazza page on Canvas.

Determination of Grades

Final Grade is based on:

20% Programming assignments
40% Midterms (20% each)
40% Final

Grade	*Points*	*Percentage*
A plus	*960 to 1000*	*96 to 100%*
A	*930 to 959*	*93 to 95%*
A minus	*900 to 929*	*90 to 92%*
B plus	*860 to 899*	*86 to 89 %*
B	*830 to 859*	*83 to 85%*
B minus	*800 to 829*	*80 to 82%*
C plus	*760 to 799*	*76 to 79%*
C	*730 to 759*	*73 to 75%*
C minus	*700 to 729*	*70 to 72%*
D plus	*660 to 699*	*66 to 69%*
D	*630 to 659*	*63 to 65%*
D minus	*600 to 629*	*60 to 62%*

Embedded Tutoring

Embedded tutoring is a form of supplemental instruction offered by Peer Connections in which the tutor attends class meetings and, under the instructor's guidance, helps students understand concepts and contribute to class discussions. The tutor also holds weekly "office hours" that students are encouraged to attend. For more information, please visit peerconnections: http://peerconnections.sjsu.edu/

Tutor name: Yvonne Hoang (Section 6)
Tutor email: yvonne.hoang@sjsu.edu
Office hours by appointment only:
2PM - 4PM Mondays and Wednesdays

Extra points: If you attend one 30' of tutoring until Midterm 2 you will receive 1%.

Other Tutoring

Appointments at main location Student Services Center 600 (SSC 600):

Tutor's Name	**Dates**	**Time**
Dev Kapupara	Mondays/Wednesdays	1 - 4pm
Dev Kapupara	Thursday	11 am – 1 pm
Yehya Abdelhadi	Tuesday	1 – 3 pm
Kevin Ngo	Mondays/Wednesday/Fridays	11 am – 1 pm

Drop-in Tutoring at Clark Hall (1st floor):

Tutor's Name	**Dates**	**Time**
Dev Kapupara	Tuesdays	11 am – 1 pm
Yehya Abdelhadi	Thursdays	1 – 3 pm

You can make appointments online (through Spartan Connect), by phone with the Welcome Desk Staff at 408-924-2587, or in-person at the main location Student Services Center 600 (SSC 600).

Tutoring takes place at our main location and we also have appointments at the Spartan Study Hub in Campus Village Building B (CVB).

Here is the link on how to schedule an appointment:

http://peerconnections.sjsu.edu/appointments/appointment/index.html

Classroom Protocol

Attendance is highly recommended. Please avoid disturbing the class: turn-off cell phones (or put them on silent or vibrate mode), no text messaging in class or during exams, **no taking pictures and video**, avoid coming late. You are not allowed to publically share or upload material for this course such as exam questions, lecture notes, or solutions without the instructor's consent.

Regrading Procedure

In the event that a student requests a regrade of a question on a homework or exam, please follow the procedure described next. A print out indicating the specific question(s) should be provided to the instructor, along with a note describing the issue and the reason for the regrading request. Please also indicate on the note your name, course section, assignment and question number. The regrading will be done by the instructor.

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' Syllabus Information web page at http://www.sjsu.edu/gup/syllabusinfo/

CS146: Data Structures and Algorithms, Fall 2019

The schedule is subject to change with fair notice.

Course Schedule

Lectures	**Date**	**Topic**	**Chapter**
1	8/21	Introduction: Algorithms & Computers, multiplication of integers	CLRS: Ch 1 & Appendix A
2	8/26	Review Data Structures (lists, stacks, queues, trees), recursion, basic sorting algorithms	CLRS: Ch 10. SW: Ch 1.3
3	8/28	Selection Sort, Insertion Sort	CLRS: Ch 2.1. SW: Ch 2.1
	9/2	Labor day	
4	9/4	Growth of functions- O, Ω, Θ, o, ω	CLRS: Ch 3. SW: Ch 1.4
5	9/9	Divide and Conquer technique: Merge Sort, Matrix Multiplication	CLRS: Ch 2.2, 2.3. SW: Ch 2.2
6	9/11	*Solving recurrences - Master Theorem & Matrix multiplication revisited*	CLRS: Ch 4.3-4.5
7	9/16	Intro to Heaps, Priority Queues	CLRS: Ch 6.1 SW: Ch 2.4
8	9/18	Heapsort	CLRS: Ch 6 SW: Ch 2.4
9	9/23	Graphs, BFS	CLRS: Appendix B.1, B.4-5, Ch 22.1
10	9/25	DFS	CLRS: Ch 22.2 SW: Ch 4.1-4.2
11	9/30	Topological sort	CLRS: Ch 22.3-5
12	10/2	SCC	CLRS: Ch 22.3-5
	10/7	**Midterm 1**	
13	10/9	Quicksort, Analysis of Quicksort	CLRS: Ch 7 (not 7.3). SW: Ch 2.3

Lectures	**Date**	**Topic**	**Chapter**
14	10/14	*Midterm review / Order statistics - Selection Algorithm*	CLRS: Ch 9 (not 9.2)
15	10/16	Sorting in linear time, Counting sort, Radix sort, Bucket sort	CLRS: Ch 8
16	10/21	Binary Search Trees	CLRS: Ch 12 SW: Ch 3.2
17	10/23	Balanced search trees: Red Black trees	CLRS: Ch 13 SW: Ch 3.3
18	10/28	*Hashing*	CLRS: Ch 11 SW: Ch 3.4
19	10/30	Union-Find: Data Structures for Disjoint Sets, Union Find, Dynamic sets	CLRS: Ch 12. SW: Ch 1.5
20	11/4	Minimum Spanning Tree (greedy) – Prim’s & Kruskal's Algorithm	CLRS: Ch 23, Ch 21 SW: Ch 4.3
21	11/6	Single Source Shortest Paths: Dijkstra’s Algorithm (greedy), Bellman-Ford introduction (dynamic)	CLRS: Ch 24 SW: Ch 4.4
	11/11	Veterans Day	
	11/13	**Midterm 2**	
22	11/18	*Midterm review / Greedy technique*	CLRS: Ch 16
23	11/20	*Greedy technique (Huffman codes, scheduling, clustering)*	CLRS: Ch 16
24	11/25	*Dynamic Programming technique (Fibonacci, Bellman-Ford again, All-Pairs Shortest Paths: Floyd-Warshall)*	CLRS: Ch 15
	11/27	*Thanksgiving non-instruction day*	
25	12/2	*Dynamic Programming (knapsack, LCS/sequence alignment, optimal search trees, independent set)*	CLRS: Ch 15
26	12/4	*NP-completeness, Reductions*	CLRS: Ch 34.1-4
27	12/9	*NP-complete problems , Review for exam*	CLRS: Ch 34.5 SW: Ch. 6.5
		Final exam **Section 6: 12/12, 9:45 – 12 noon** **Section 8: 12/16, 14:45 – 17**	