San José State University
Computer Science Department

CS146, Sections 2 and 3, Data Structures and Algorithms, Spring 2020

Course and Contact Information

Instructor:	William “Bill” Andreopoulos
Office Location:	MacQuarrie Hall 416
Email:	william.andreopoulos@sjsu.edu Please use Canvas Messaging and the Discussion Forum
Office Hours:	Monday 16:00-17:00 pm and Wednesday 16:00-17:00
Class Days/Time:	MW 9:00-10:15 am (Section 2) and 10:30-11:45 am (Section 3)
Classroom:	MacQuarrie Hall 223 (Sections 2 and 3)
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.

Readings and pre-class assignments: This is a challenging course and we will cover a lot of material. It is recommended for students to come to class with a good idea of the material we will cover. Readings will regularly be assigned for the next class (see schedule). Slides will be posted under the Canvas modules before the next class. Screencasts covering the slides will also be placed under Canvas in the form of short videos. Students are expected to have reviewed the slides and screencasts before the next class. By viewing the screencasts and slides before class, students will be able to bring their questions and keep up with the pace in the classroom.

In-class worksheets: there will be in-class problem solving, to be done with the assistance of the embedded tutor. These will generally be coding problems (in Java) from the reading assignment and/or the homework. The in-class worksheets are a tool for you to learn the material, prepare for exams and practice coding for your future job interviews. These can be done in the eclipse IDE and submitted on Canvas. It is recommended to bring a laptop to class.

The worksheets are graded based on effort and get graded "complete" if a reasonable solution is proposed for each problem, even if it is incorrect (it is understood that the solution might be imperfect). Worksheet submissions are due one week after the class. Please submit what you have by the due date.

We will take time at the beginning of each class to discuss any difficulties students have in completing the worksheets from previous classes. We will also do code reviews.

While it is fine to discuss the worksheet solutions with other people, code solutions submitted on Canvas should reflect the students' own efforts in writing the code. *Do not write the code for anyone else. Do not copy code from another source, such as a website, since Canvas automatically checks submissions for plagiarism from multiple online sources.*

Quizzes: will be posted on Canvas. The quizzes are in the form of multiple choice and true-false questions. The quizzes stay open throughout the semester and students can resubmit them multiple times. Students can repeat the quizzes anytime for practicing the concepts and material. The maximum grade achieved on a quiz is the one kept. Quiz answers will also be reviewed in class with iClicker. Please install iClicker on your phone (app) or laptop (iclicker.com) following these instructions: http://www.sjsu.edu/ecampus/teaching-tools/iclicker/

Homework assignments: Programming assignments will be assigned. *Homework assignments are to be 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, true/false, short answer questions and short coding questions. Exams are *closed book*, final exam is comprehensive. Students are allowed to use a one-page sheet of notes (hand-written or printed) as an aid. No make-up exams, except in case of verifiable emergency circumstances.

Discussion Forum on Canvas

This term we will be using the Discussion Forum on Canvas 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 the Discussion Forum on Canvas.

Extra credit opportunity

A student can volunteer to present in-class her solution for an assignment or a worksheet. Students have to express interest in presenting (by messaging or speaking with the instructor). An assignment or hands-on can only be reviewed once. A review lasts for 20 minutes max. These will take the form of code reviews, where the student walks us through her code solution and we discuss the proposed solution and if there are better ways to solve the problem. Extra credit of 1% for a student who reviews her solution for an assignment or a worksheet in class.

Determination of Grades

Final Grade is based on:

30%	Assignments
30%	Midterms (15% each)
30%	Final
9%	In-class worksheets
1%	Quizzes

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%*

Communication with the instructor

Students should post any general questions on the Discussion Forum, where the entire class can benefit from the responses. To email the instructor, please use Canvas messaging rather than a direct email address, since this helps the instructor to organize and keep track of all course-related electronic communication. The instructor responds to course-related electronic messages in this priority: Discussion Forum, then messages sent through Canvas.

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: Jonathan Tshimpaka (Section 2)
Tutor email: jonathankabongo.tshimpaka@sjsu.edu
Tutoring hours by appointment only:
Monday and Wednesday 11:00am-13:00pm.

Tutor name: Yvonne Hoang (Section 3)
Tutor email: yvonne.hoang@sjsu.edu
Tutoring hours by appointment only:
Tuesday and Thursday 14:00pm-16:00pm, Friday 11:00am-13:00pm.

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	Mon/Tues/Wed/Thurs	11am -1pm
Dev Kapupara	Mondays/Wednesdays	1:30pm -2:30pm
Yehya Abdelhadi	Tuesdays	11am -1pm
Yehya Abdelhadi	Thursdays	11:30am -1:30pm
Kevin Ngo	Tuesdays	1pm -3pm
Kevin Ngo	Fridays	10am -12pm
Jonathan Tshimpaka	Mondays/Wednesdays	11am -1pm

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

Tutor's Name	Dates	Time
Dev Kapupara	Wednesdays	11 am – 1 pm
Yehya Abdelhadi	Wednesdays	10am – 12pm

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 (not email) 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.

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, Spring 2020

The schedule is subject to change with fair notice.

Course Schedule

Lectures	Date	Topic	Chapter
1	1/27	Introduction: Algorithms & Computers, multiplication of integers	CLRS: Ch 1 & Appendix A
2	1/29	Review Data Structures (lists, stacks, queues, trees), recursion, basic sorting algorithms	CLRS: Ch 10. SW: Ch 1.3
3	2/03	Selection Sort, Insertion Sort	CLRS: Ch 2.1. SW: Ch 2.1
4	2/05	Growth of functions- O, Ω, Θ, o, ω	CLRS: Ch 3. SW: Ch 1.4
5	2/10	Divide and Conquer technique: Merge Sort, Matrix Multiplication	CLRS: Ch 2.2, 2.3. SW: Ch 2.2
6	2/12	Solving recurrences - Master Theorem & Matrix multiplication revisited	CLRS: Ch 4.3-4.5
7	2/17	Intro to Heaps, Priority Queues	CLRS: Ch 6.1 SW: Ch 2.4
8	2/19	Heapsort	CLRS: Ch 6 SW: Ch 2.4
9	2/24	Graphs, BFS	CLRS: Appendix B.1, B.4-5, Ch 22.1
10	2/26	DFS	CLRS: Ch 22.2 SW: Ch 4.1-4.2
11	3/02	Topological sort	CLRS: Ch 22.3-5
12	3/04	SCC	CLRS: Ch 22.3-5
	3/09	**Midterm 1**	
13	3/11	Quicksort, Analysis of Quicksort	CLRS: Ch 7 (not 7.3). SW: Ch 2.3

Lectures	Date	Topic	Chapter
14	3/16	*Midterm review* / Order statistics - Selection Algorithm	CLRS: Ch 9 (not 9.2)
15	3/18	Sorting in linear time, Counting sort, Radix sort, Bucket sort	CLRS: Ch 8
16	3/23	Binary Search Trees	CLRS: Ch 12 SW: Ch 3.2
17	3/25	Balanced search trees: 2-3 trees	CLRS: Ch 13 SW: Ch 3.3
	3/30	*Spring recess*	
	4/01	*Spring recess*	
18	4/06	Hashing	CLRS: Ch 11 SW: Ch 3.4
19	4/08	Union-Find: Data Structures for Disjoint Sets, Union Find, Dynamic sets	CLRS: Ch 12. SW: Ch 1.5
20	4/13	Minimum Spanning Tree (greedy) – Prim's & Kruskal's Algorithm	CLRS: Ch 23, Ch 21 SW: Ch 4.3
21	4/15	Single Source Shortest Paths: Dijkstra's Algorithm (greedy), Bellman-Ford introduction (dynamic)	CLRS: Ch 24 SW: Ch 4.4
	4/20	**Midterm 2**	
22	4/22	*Midterm review* / Greedy technique	CLRS: Ch 16
23	4/27	Greedy technique (Activity Selection, knapsack, Huffman codes, scheduling, clustering)	CLRS: Ch 16
24	4/29	Dynamic Programming technique (Activity Selection, Fibonacci, Bellman-Ford again, All-Pairs Shortest Paths: Floyd-Warshall)	CLRS: Ch 15
25	5/04	Dynamic Programming (Knapsack, LCS/sequence alignment, optimal search trees, independent set)	CLRS: Ch 15
26	5/06	NP-completeness, Reductions	CLRS: Ch 34.1-4
27	5/11	NP-complete problems , Review for exam	CLRS: Ch 34.5 SW: Ch. 6.5
		Final exam **Section 2:** Monday, May 18, 07:15-09:30 **Section 3:** Friday, May 15, 09:45-12:00	