# San José State University
# College of Science/Department of Computer Science
# CS146, Data Structures and Algorithms, Section 7, Spring, 2020

**Course and Contact Information**

| | |
|---|---|
| Instructor: | Aniket Chandak |
| Office Location: | DH 282 |
| Email: | aniket.chandak@sjsu.edu |
| Office Hours: | Tu/Th 3:15 PM – 4:15 PM |
| Class Days/Time: | Tu/Th 4:30-5:45pm |
| Classroom: | SH 100 |

**Course Format**

**Faculty Web Page and MYSJSU Messaging**

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on my faculty web page at http://www.sjsu.edu/people/firstname.lastname and/or 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.

**Prerequisite**

MATH 42, MATH 30, CS 46B, and CS 49J (or equivalent knowledge of Java) (with a grade of "C-" or better in each); Computer Science, Applied and Computational Math or Software Engineering majors only; or instructor consent

**Course Description**

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

**Course Objectives**

1. To ensure that students are familiar with ways to implement elementary data structures and their associated algorithms.
2. To introduce students to the implementation of more complex data structures and their associated algorithms.
3. To acquaint students with advanced sorting techniques.
4. To teach students how to determine the time complexity of algorithms.
5. To introduce students to algorithm design techniques.

**Course Learning Outcomes (CLO)**

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

1. 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. Prove basic properties of trees and graphs
3. Perform breadth-first search and depth-first search on directed as well as undirected graphs
4. Use advanced sorting techniques (heapsort, mergesort, quicksort)
5. Determine the running time of an algorithm in terms of asymptotic notation
6. Solve recurrence relations representing the running time of an algorithm designed using a divide-and-conquer strategy
7. 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
8. Understand algorithms designed using greedy, divide-and-conquer, and dynamic programming techniques

## Required Texts/Readings

### Textbook

Recommended reading:

Algorithms, 4th Edition by Robert Sedgewick and Kevin Wayne

Addison-Wesley Professional, 2011, ISBN 0-321-57351-X.

https://www.amazon.com/dp/032157351X/

Introduction to Algorithms, 3rd Edition Cormen, Leiserson, Rivest, and Stein
ISBN-10: 0262033844ISBN-13: 978-0262033848MIT Press, 2009
https://www.amazon.com/Introduction-Algorithms-3rd-MIT-Press/dp/0262033844
You can find errata (bug reports) for the book http://www.cs.dartmouth.edu/~thc/clrs-bugs/bugs-3e.php,for whichever printing of the book you get.

Optional reading:

Introduction to Programming in Java: An Interdisciplinary Approach, 1st Edition by Robert Sedgewick and Kevin Wayne

ISBN-13: 978-0321498052

https://www.amazon.com/gp/product/0321498054/

## Programming Environment

Please Follow the instructions on the Author's website: http://algs4.cs.princeton.edu/home/

## 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. More details about student workload can be found in University Policy S12-3at http://www.sjsu.edu/senate/docs/S12-3.pdf.

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

**Midterm exams:**

There will be two Midterm exams during the semester.

**Final Examination:**

One written final cumulative exam.
Section 7: Day: May 13 Time: 14:45-17:00

The exams will contain multiple choice questions, short answer questions and questions that require pseudocode and/or computations. Students must obtain >50% in each component of the course (homework, project, quizzes & written exams) in order to be eligible for a passing grade.

**Grading Information**
Your grade for the course will be based on the following components:

- Mid Term Exams - 30%
- Final Exam - 25 %
- Projects 20 %
- Assignments 20%
- Discretion - 5% (Including participations in class, answering forum posts on Piazza)

Exams are closed book; final exam is comprehensive. No extra point options in the final. No make-ups exams except in case of verifiable emergency circumstances

**Determination of Grades**
The following shows the grading scale to be used to determine the letter grade:

| **Percentage** | **Grade** |
|---|---|
| 95 and above | A+ |
| 92-94 | A |
| 90 - 91 | A- |
| 87 - 89 | B+ |
| 83 - 86 | B |
| 80 - 82 | B- |
| 77 - 79 | C+ |
| 73 - 76 | C |
| 70 - 72 | C- |
| 67 - 69 | D+ |
| 63-66 | D |
| 60-62 | D- |
| 59 and below | F |

**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. You may not publicly share or upload material for this course such as exam questions, lecture notes, or solutions without my consent.

**University Policies (Required)**

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/" Make sure to review these policies and resources.

**Policies**

Late homework/projects

All homework and projects will be due on Thursday of the indicated week at 5:00pm. The assignment will be posted at least a week before the due date to give enough time to work, and ask for help during my office hours or online on Piazza. Please do not email me few hours before the deadline asking me to help you understand concepts. If I feel that you just start working on your assignment at the due date, I will ignore your emails because I know you won't be able to finish understanding the problem, coding, testing, compiling, and debugging in a few hours. So, please start early and manage your time wisely.

Late Submission: (max one assignment is ok)

- 0-6hr -> no penalty
- 6-12hr -> 30% penalty
- 12hr- 24hr ->60%
- +2 Day -> 100%

Classroom Etiquette

Please come to class on time and if you have to leave please do so quietly. It is fine to eat and drink during the lecture as long as it does not disturb your classmates and you are doing it quietly. Refrain from crunchy or smelly food (unless you bring enough for everyone else!). If you are not a surgeon who is on call, try to put your phone on silent mode (not vibrate mode). Abstain from talking to each other during class and if you have any question or concern, please bring them up to me.

Piazza Etiquette

Ask clear questions to get better answers. Make sure your question has not been already asked and answered. Read these guidelines (http://superuser.com/help/how-to-ask) when you ask a question. It is fine to share a couple of lines of code but please don't just post your code or share a significant amount of code. Try to ask general questions.

Piazza sign up link: piazza.com/sjsu/spring2020/cs146

**Tips to succeed**

- This is a challenging, programing-intensive course. It gets more difficult very quickly as the semester progress. It is not about coding, but it teaches you how to write an "efficient" and "elegant" code. So, please do not fall behind.
- Ask for help. I want you to be successful in this course and I love questions. It's important to ask questions during the class. This is your chance to understand the materials before you go home and read it on your own. There is no stupid question and always know that other students will benefit from the questions you ask during class.
- Do not get frustrated. Programming takes patience. It is very common to spend hours doing your HWs/projects. Take short breaks. If you spend hours on one bug or an error, email me and I will help you. Always remember, the more you

code, the better you will become.

- Always keep testing your code. After ever few lines of coding, test and compile your code. It is easy to find mistakes in a small scope. And remember to back up your code while you are working on it. I recommend saving it in the cloud.
- As the course is challenging, and required a significant time commitment for most students, ask your instructor for help, ask your friends on Piazza, but there is no excuse for plagiarism.

# CS146, Data Structures and Algorithms, Section 7, Spring, 2020

**Course Schedule**

| Week | Date | Topics, Readings, Assignments, Deadlines |
|---|---|---|
| 1 | 1/23 | Introduction, Course mechanic & Logistics |
| 1 | 1/28 | Programming Model (Reading: SW 1.1) |
| 2 | 1/30 | Data Abstraction (Reading: SW 1.2) |
| 2 | 2/4 (drop) | Basic Data Structures (Reading: SW 1.3) |
| 3 | 2/6 | Cont. Basic Data Structures (Reading: SW 1.3) |
| 3 | 2/11 (add) | Analysis of Algorithms, Growth of functions-O, Ω, Θ, ο, ω (Reading: CLRS Ch 3; SW 1.4) |
| 4 | 2/13 | Cont. Analysis of Algorithms, Growth of functions-O, Ω, Θ, ο, ω (Reading: CLRS Ch 3; SW 1.4) |
| 4 | 2/18 | Union-find (Reading: SW 1.5) |
| 5 | 2/20 | Cont. Union-find (Reading: SW 1.5) |
| 5 | 2/25 | Elementary Sorts (Reading: SW 2.1) |
| 6 | 2/27 | Quick Sort & Divide and Conquer technique: Merge Sort (Reading: SW 2.2, 2.3) |
| 6 | 3/3 | Priority Queues (Reading: SW 2.4) |
| 7 | 3/5 | Cont. Priority Queues (Reading: SW 2.4) |
| 7 | 3/10 | Sorting Applications (Reading: SW 2.5) |
| 8 | 3/12 | Symbol Tables & Binary Search Trees (Reading: SW 3.1, 3.2) |
| 8 | 3/17 | Balanced Search Trees 2-3 search trees (Reading: SW 3.3) |
| 9 | 3/19 | **Midterm I** |
| 9 | 3/24 | Cont. Balanced Search Trees Red-black BSTs (Reading: SW 3.3) |
| 10 | 3/26 | Hashing (Reading: SW 3.4) |
|  | 3/31 – 4/2 | *Spring Break* |
| 10 | 4/7 | Searching Applications (Reading: SW 3.5) |
| 11 | 4/9 | Undirected Graphs (Reading: SW 4.1) |
| 11 | 4/14 | Cont. Undirected Graphs (Reading: SW 4.2) |
| 12 | 4/16 | Directed Graphs (Reading: SW 4.2) |
| 12 | 4/21 | Minimum Spanning Trees, Prim and Kruskal algorithms, Data Structures for Disjoint Sets (Reading: CLRS Ch 23, Ch 21; SW 4.3) |
| 13 | 4/23 | **Midterm II** |
| 13 | 4/28 | Shortest Paths: Dijkstra's algorithm and Bellman-Ford (Reading: CLRS Ch 24; SW 4.4) |
| 14 | 4/30 | Greedy technique (Reading: CLRS Ch 16) |
| 14 | 5/5 | Dynamic Programming technique (Reading: CLRS Ch 15) |
| 15 | 5/7 | Dynamic Programming technique (Reading: CLRS Ch 15) |
| 15 | 5/13 | Final Exam |