COLORADO STATE UNIVERSITY
GLOBAL

CSC506: DESIGN AND ANALYSIS OF ALGORITHMS

Credit Hours: 3

Contact Hours: This is a 3-credit course, offered in accelerated format. This means that 16 weeks of material is covered in 8 weeks. The exact number of hours per week that you can expect to spend on each course will vary based upon the weekly coursework, as well as your study style and preferences. You should plan to spend 14-20 hours per week in each course reading material, interacting on the discussion boards, writing papers, completing projects, and doing research.

Faculty Information: Faculty contact information and office hours can be found on the faculty profile page.

COURSE DESCRIPTION AND OUTCOMES

Course Description:

This graduate course provides students with a foundational knowledge in the design and analysis of algorithms. Students will make use of appropriate data structures. Study of the complexity and analysis of algorithms will be completed focusing on worst case and average case, lower bounds, NP-completeness, and recurrences. Students will explore the complexity of appropriate searching, sorting, and graphing algorithms.

Course Overview:

Algorithms and data structures are the most fundamental concepts in computing. They are the building blocks from which complex software is built. Algorithms, in their simplest form, are just a sequence of actions, a list of instructions. Essentially algorithms are composed of four elements, all of which we will discuss in this course: sequential operations, actions based on the state of a data structure, iteration, and recursion. You will also explore many other algorithm concepts including lower bounds, abstract data types, graphing algorithms, and Big-O Notation.

Course Learning Outcomes:

1. Identify factors that can affect the lower bound of a solution.
2. Discuss the use of abstract data types in software development
3. Create an application that demonstrates optimal performance.
4. Implement a recursive solution to solve a specific problem.
5. Develop an application that makes use of appropriate data structures
6. Evaluate the Big-O runtime of an algorithm.

Participation & Attendance

Prompt and consistent attendance in your online courses is essential for your success at CSU-Global Campus. Failure to verify your attendance within the first 7 days of this course may result in your withdrawal. If for some reason you would like to drop a course, please contact your advisor.

Online classes have deadlines, assignments, and participation requirements just like on-campus classes. Budget your time carefully and keep an open line of communication with your instructor. If you are having technical problems, problems with your assignments, or other problems that are impeding your progress, let your instructor know as soon as possible.

Course Materials

Required:

CSC506: Design and Analysis of Algorithms zybook (learn.zybooks.com/library) ISBN: 9781394012268

Zybook code: CSC506ZybookInteractiveText2020

NOTE: *All non-textbook required readings and materials necessary to complete assignments, discussions, and/or supplemental or required exercises are provided within the course itself. Please read through each course module carefully.*

Course Schedule

Due Dates

The Academic Week at CSU Global begins on Monday and ends the following Sunday.

- **Discussion Boards:** The original post must be completed by Thursday at 11:59 p.m. MT and Peer Responses posted by Sunday 11:59 p.m. MT. Late posts may not be awarded points.
- **Critical Thinking:** Assignments are due Sunday at 11:59 p.m. MT.

Weekly Reading and Assignment Details

Module 1

Readings

- Chapter 1 in *CSC506: Design and Analysis of Algorithms*
- Tutorialspoint. (2020). Data structures & algorithms – overview. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/data_structure_overview.htm
- Tutorialspoint. (2020). Data structures – algorithms basics. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/algorithms_basics.htm

Discussion (25 points)

Critical Thinking (75 points)

OPTION #1: Algorithms in Practice
Discuss what an algorithm is and how you perceive it. Then, write an algorithm for each of the following tasks:

a. Making a peanut butter and jelly sandwich
b. Getting up in the morning
c. Doing your homework
d. Driving home in the afternoon
e. Waiting in line at the grocery store

Your paper should be 3-4 pages in length (not including title and references pages) and conform to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

OPTION #2: Algorithms in Practice
Discuss what an algorithm is and how you perceive it. Then, design an algorithm that, when given an arrangement of the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, rearranges the digits so that the new arrangement represents the next larger value that can be represented by these digits (or reports that no such rearrangement exists if no rearrangement produces a larger value). Thus, 5647382901 would produce 5647382910.

Your paper should be 3-4 pages in length (not including title and references pages) and conform to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Module 2

Readings

- Chapter 2 in *CSC506: Design and Analysis of Algorithms*
- Tutorialspoint. (2020). Data structure and algorithms linear search. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/linear_search_algorithm.htm
- Tutorialspoint. (2020). Data structure and algorithms binary search. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/binary_search_algorithm.htm
- Debnath, S. (n.d.). Analysis of algorithms: Big-O analysis. Retrieved from https://www.geeksforgeeks.org/analysis-algorithms-big-o-analysis/

Discussion (25 points)

Critical Thinking (75 points)

OPTION #1: String Edit Distance Problem
Suppose that you want to transform the word “algorithm” into the word “alligator.” For each letter you can either copy the letter from one word to another at a cost of 5, you can delete a letter at cost of 20, or insert a letter at a cost of 20. The total cost to transform one word into another is used by spell check programs to provide suggestions for words that are close to one another. Use dynamic programming techniques to develop an algorithm that gives you the smallest edit distance between any two words.

Your Python Programming submission materials must include your source code and screenshots of the Python interface executing the application and the results.

Additionally, you must include one paragraph describing how you completed this activity or where you had trouble executing the code.

OPTION #2: Algorithm Analysis
Develop two algorithms, one based on a loop structure and the other on a recursive structure, to print the daily salary of a worker who each day is paid twice the previous day's salary (starting with one penny for the first day's work) for a 30-day period.

What problems relating to number storage are you likely to encounter if you implement your solutions on an actual machine?

Your paper should be 3-4 pages in length (not including title and references pages) and conform to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Module 3

Readings

- Chapter 3 in *CSC506: Design and Analysis of Algorithms*
- Tutorialspoint. (2020). Data structure – sorting techniques. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/sorting_algorithms.htm
- Tutorialspoint. (2020). Data structure and algorithms insertion sort. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/insertion_sort_algorithm.htm
- Tutorialspoint. (2020). Data structures – merge sort algorithm. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/merge_sort_algorithm.htm
- Tutorialspoint. (2020). Data structure and algorithms – quick sort. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/quick_sort_algorithm.htm

Discussion (25 points)

Critical Thinking (75 points)

OPTION #1: Random Number Generator
Using a random number generator, create a list of 500 integers. Perform a benchmark analysis using some of the sorting algorithms from this module. What is the difference in execution speed between the different sorting algorithms? In your paper, be sure to provide a brief discussion of the sorting algorithms used in this activity.

Your paper should be 2-3 pages in length and conform to *CSU Global Guide to Writing and APA*. Include at least one scholarly reference in addition to the course textbook. The CSU Global Library is a good place to find these references.

OPTION #2: Minimum Number
Write two Python functions to find the minimum number in a list. The first function should compare each number to every other number on the list $O(n^2)$. The second function should be linear $O(n)$.

Briefly discuss how you created your activity and what problems you may have encountered. Additionally, discuss whether the algorithm could have been implemented a different way.

Your paper should be 2-3 pages in length and conform to *CSU Global Guide to Writing and APA*. Include at least one scholarly reference in addition to the course textbook. The CSU Global Library is a good place to find these references.

Portfolio Milestone (50 points)

OPTION #1: Analysis of Algorithms and Data Structures

Milestone 1: Topic Selection and Outline

For Milestone 1, you will complete an outline of the portfolio project. Include an outline of your planned analyses and expected results. This milestone only requires high-level ideas and organization. The goal of this milestone is to ensure you are headed in the right direction to complete the final project successfully.

Your paper should be 1-3 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

For your reference: Portfolio Project Scenario

Your portfolio project is to analyze the pragmatic differences in how two or more different algorithms and/or data structures solve the same problem. You will choose the subject of your project, code and profile the algorithms in Python programming language, and analyze your results. You will then present your findings in a well-organized research paper that resembles the structure of a manuscript for publication in a computer science journal.

Your paper should be 8-12 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least six scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

OPTION #2: Retroactive Search Trees

Milestone 1: Topic Selection and Outline

In the portfolio project, retroactive search trees will be implemented - both partially and fully retroactive search trees. The update operations to the (non-retroactive) search tree should be Insert(x) and Delete(x), and the query operation should be Pred(x) that returns the largest element stored in the subtree ≤x.

The task of Milestone 1 is to:

- Define an appropriate interface to a partially and a fully retroactive search tree.

Your paper should be 1-3 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

As a reminder: Portfolio Project Scenario

In this portfolio project, retroactive search trees will be implemented - both partially and fully retroactive search trees. The update operations to the (non-retroactive) search tree should be Insert(x) and Delete(x), and the query operation should be Pred(x) that returns the largest element stored in the subtree ≤x. The tasks of the project are to:

- Define an appropriate interface to a partially and a fully retroactive search tree.
- Implement a search tree, a partially retroactive search tree, and a fully retroactive search tree.
- Test if your retroactive solutions are correct by comparing them with simple rollback solutions.

Compare the performance of the different data structures. Compare the performance of the fully retroactive search tree with a simple rollback solution. What are the thresholds where the complicated solution is superior to the rollback solution?

You will present your findings in a well-organized research paper that resembles the structure of a manuscript for publication in a computer science journal.

Your paper should be 8-12 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least six scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Module 4

Readings

- Chapter 4 in *CSC506: Design and Analysis of Algorithms*
- Tutorialspoint. (2020). Data structure and algorithms – stack. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/stack_algorithm.htm
- Tutorialspoint. (2020). Data structure and algorithms – queue. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/dsa_queue.htm
- Tutorialspoint. (2020). Data structure and algorithms – linked list. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/linked_list_algorithms.htm
- Tutorialspoint. (2020). Data structure – doubly linked list. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/doubly_linked_list_algorithm.htm

Discussion (25 points)

Critical Thinking (75 points)

OPTION #1: List Based Stack
Design and implement an experiment that will compare the performance of the Python list based stack and queue with the linked list implementation. Provide a brief discussion of both stacks and queues for this activity.

Your Python Programming submission materials must include your source code and screenshots of the Python interface executing the application and the results.

Your paper should be 1-2 pages in length and conform to *CSU Global Guide to Writing and APA*. Include at least one scholarly reference in addition to the course textbook. The CSU Global Library is a good place to find these references.

OPTION #2: Unordered and Ordered Lists
Consider the relationship between Unordered and Ordered lists. Is it possible that inheritance could be used to build a more efficient implementation? Implement this inheritance hierarchy. Provide the screenshot of your inheritance hierarchy in your paper.

Your paper should be 2-3 pages in length and conform to *CSU Global Guide to Writing and APA*. Include at least one scholarly reference in addition to the course textbook. The CSU Global Library is a good place to find these references.

Module 5

Readings

- Chapter 5 in *CSC506: Design and Analysis of Algorithms*
- Tutorialspoint. (2020). Data structure and algorithms – hash table. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/hash_data_structure.htm
- Tutorialspoint. (2020). Heap data structures. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/heap_data_structure.htm
- GeeksforGeeks. (n.d.). Treap (a randomized binary search tree). Retrieved from https://www.geeksforgeeks.org/treap-a-randomized-binary-search-tree/

Discussion (25 points)

Critical Thinking: Title (75 points)

OPTION #1: Linear Probing
Illustrate the linear probing method in hashing. Explain its performance analysis. Lastly, discuss how rehashing overcomes the drawbacks of linear probing. Provide at least one visual in your activity.

Your paper should be 2-3 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

OPTION #2: Heaps
What are the properties of a heap? What is the running time of heap sort on an array A of length n that is already sorted in increasing order? What about the same in decreasing order? Lastly, trace out the heap sort algorithm for the following list: {25, 44, 55, 99, 30, 37, 15, 10, 2, 4}.

Your activity should include the following: 1. A description of the algorithm(s) and, if helpful, pseudocode. 2. At least one worked example or diagram to show more precisely how the algorithm(s) works.

Your paper should be 2-3 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least one scholarly resource in addition to the course textbook. The CSU Global Library is a good place to find resources.

Module 6

Readings

- Chapter 6 in *CSC506: Design and Analysis of Algorithms*
- Tutorialspoint. (2020). Data structure and algorithms – tree. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/tree_data_structure.htm
- Tutorialspoint. (2020). Data structure – binary search tree. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/binary_search_tree.htm
- Tutorialspoint. (2020). Data structure and algorithms – AVL trees. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/avl_tree_algorithm.htm

Discussion (25 points)

Critical Thinking (75 points)

OPTION #1: Binary Search Tree

You'll build a simple binary search tree in this activity.

- Build a Node class. It is should have attributes for the data it stores as well as its left and right children. As a bonus, try including the Comparable module and make nodes compare using their data attribute.
- Build a Tree class which accepts an array when initialized. The Tree class should have a root attribute which uses the return value of #build_tree which you'll write next.
- Write a #build_tree method which takes an array of data (e.g. [1, 7, 4, 23, 8, 9, 4, 3, 5, 7, 9, 67, 6345, 324]) and turns it into a balanced binary tree full of Node objects appropriately placed (don't forget to sort and remove duplicates!). The #build_tree method should return the level-1 root node.
- Write an #insert and #delete method which accepts a value to insert/delete.

Compile and submit your source code and screenshots of the application executing the application and the results in a single document.

OPTION #2: Driver Script

Write a driver script that does the following:

1. Create a binary search tree from an array of random numbers (`Array.new(15) { rand(1..100) }`)
2. Confirm that the tree is balanced by calling `#balanced?`
3. Print out all elements in level, pre, post, and in order
4. try to unbalance the tree by adding several numbers > 100
5. Confirm that the tree is unbalanced by calling `#balanced?`
6. Balance the tree by calling `#rebalance!`
7. Confirm that the tree is balanced by calling `#balanced?`
8. Print out all elements in level, pre, post, and in order

Compile and submit your source code and screenshots of the application executing the application and the results in a single document.

Portfolio Milestone (50 points)

OPTION #1: Analysis of Algorithms and Data Structures

Milestone 2: First Draft

For Milestone 2, you will complete a rough draft of the portfolio project. At a minimum, background material should be complete, and you should have at least one set of benchmark results finished. Ideally, you will have completed most of your technical work for Milestone 2.

Your paper should be 4-6 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least four scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

As a reminder: Portfolio Project Scenario

Your portfolio project is to analyze the pragmatic differences in how two or more different algorithms and/or data structures solve the same problem. You will choose the subject of your project, code and profile the algorithms in the Python programming language, and analyze your results. You will then present your findings in a well-organized research paper that resembles the structure of a manuscript for publication in a computer science journal.

Your paper should be 8-12 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least six scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

OPTION #2: Retroactive Search Trees

Milestone 2: First Draft

In the portfolio project, retroactive search trees will be implemented - both partially and fully retroactive search trees. The update operations to the (non-retroactive) search tree should be Insert(x) and Delete(x), and the query operation should be Pred(x) that returns the largest element stored in the subtree ≤x.

The tasks of Milestone 2 are to:

- Implement a search tree, a partially retroactive search tree, and a fully retroactive search tree.
- Test if your retroactive solutions are correct by comparing them with simple rollback solutions.

Your paper should be 1-3 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

As a reminder: Portfolio Project Scenario

In this portfolio project, retroactive search trees will be implemented - both partially and fully retroactive search trees. The update operations to the (non-retroactive) search tree should be Insert(x) and Delete(x), and the query operation should be Pred(x) that returns the largest element stored in the subtree ≤x. The tasks of the project are to:

- Define an appropriate interface to a partially and a fully retroactive search tree.
- Implement a search tree, a partially retroactive search tree, and a fully retroactive search tree.
- Test if your retroactive solutions are correct by comparing them with simple rollback solutions.

Compare the performance of the different data structures. Compare the performance of the fully retroactive search tree with a simple rollback solution. What are the thresholds where the complicated solution is superior to the rollback solution?

You will present your findings in a well-organized research paper that resembles the structure of a manuscript for publication in a computer science journal.

Your paper should be 8-12 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least six scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Module 7

Readings

- Chapter 7 in *CSC506: Design and Analysis of Algorithms*
- Tutorialspoint. (2020). Data structure – graph data structure. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/graph_data_structure.htm
- Rao, K.P.K., & Murugan, T.S. (2019). An efficient routing algorithm for software defined networking using Bellman Ford Aagorithm. International Journal of Online and Biomedical Engineering, (14), 87. Retrieved from https://www.researchgate.net/publication/336833451_An_Efficient_Routing_Algorithm_for_Software_Defined_Networking_using_Bellman_Ford_Algorithm

Discussion (25 points)

Module 8

Readings

- Chapter 8 in *CSC506: Design and Analysis of Algorithms*
- Tutorialspoint. (2020). Data structure – bubble sort algorithm. Retrieved from https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm
- Ghughtyal, S. (n.d.). Python set operations (union, intersection, difference and symmetric difference). Retrieved from https://www.geeksforgeeks.org/python-set-operations-union-intersection-difference-symmetric-difference/

Discussion (25 points)

Portfolio Project (250 points)

OPTION #1: Analysis of Algorithms and Data Structures
Your portfolio project is to analyze the pragmatic differences in how two or more different algorithms and/or data structures solve the same problem. You will choose the subject of your project, code and profile the algorithms in the Python programming language, and analyze your results. You will then present your findings in a well-organized research paper that resembles the structure of a manuscript for publication in a computer science journal.

Your paper should be 8-12 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least six scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

OPTION #2: Retroactive Search Trees
In this portfolio project, retroactive search trees will be implemented - both partially and fully retroactive search trees. The update operations to the (non-retroactive) search tree should be Insert(x) and Delete(x), and the query operation should be Pred(x) that returns the largest element stored in the subtree ≤x. The tasks of the project are to:

- Define an appropriate interface to a partially and a fully retroactive search tree.
- Implement a search tree, a partially retroactive search tree, and a fully retroactive search tree.
- Test if your retroactive solutions are correct by comparing them with simple rollback solutions.

Compare the performance of the different data structures. Compare the performance of the fully retroactive search tree with a simple rollback solution. What are the thresholds where the complicated solution is superior to the rollback solution?

You will present your findings in a well-organized research paper that resembles the structure of a manuscript for publication in a computer science journal.

Your paper should be 8-12 pages in length, not including title and references pages, and conform to *CSU Global Guide to Writing and APA*. Include at least six scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

COURSE POLICIES

Grading Scale	
A	95.0 – 100
A-	90.0 – 94.9
B+	86.7 – 89.9
B	83.3 – 86.6
B-	80.0 – 83.2
C+	75.0 – 79.9
C	70.0 – 74.9
D	60.0 – 69.9
F	59.9 or below

Course Grading

20% Discussion Participation
45% Critical Thinking Assignments
35% Final Portfolio Project
0% Live Classroom

In-Classroom Policies

For information on late work and incomplete grade policies, please refer to our **In-Classroom Student Policies and Guidelines** or the Academic Catalog for comprehensive documentation of CSU-Global institutional policies.

Academic Integrity
Students must assume responsibility for maintaining honesty in all work submitted for credit and in any other work designated by the instructor of the course. Academic dishonesty includes cheating, fabrication, facilitating academic dishonesty, plagiarism, reusing /re-purposing your own work (see *CSU-Global Guide to Writing and APA Requirements* for percentage of repurposed work that can be used in an assignment), unauthorized possession of academic materials, and unauthorized collaboration. The CSU-Global Library provides information on how students can avoid plagiarism by understanding what it is and how to use the Library and Internet resources.

Citing Sources with APA Style
All students are expected to follow the *CSU-Global Guide to Writing and APA Requirements* when citing in APA (based on the APA Style Manual, 6th edition) for all assignments. For details on CSU-Global APA style, please review the APA resources within the CSU-Global Library under the “APA Guide & Resources” link. A link to this document should also be provided within most assignment descriptions in your course.

Disability Services Statement
CSU–Global is committed to providing reasonable accommodations for all persons with disabilities. Any student with a documented disability requesting academic accommodations should contact the Disability Resource Coordinator at 720-279-0650 and/or email ada@CSUGlobal.edu for additional information to coordinate reasonable accommodations for students with documented disabilities.

Netiquette
Respect the diversity of opinions among the instructor and classmates and engage with them in a courteous, respectful, and professional manner. All posts and classroom communication must be conducted in accordance with the student code of conduct. Think before you push the Send button. Did you say just what you meant? How will the person on the other end read the words?

Maintain an environment free of harassment, stalking, threats, abuse, insults or humiliation toward the instructor and classmates. This includes, but is not limited to, demeaning written or oral comments of an ethnic, religious, age, disability, sexist (or sexual orientation), or racist nature; and the unwanted sexual advances or intimidations by email, or on discussion boards and other postings within or connected to the online classroom. If you have concerns about something that has been said, please let your instructor know.