Data Structures & Algorithms

Fall 2024

Tentative Syllabus, subject to change
12 credits
Course Catalog Entry

Faculty

Paul Pham

Weekly Schedule

	Monday	Tuesday	Wednesday	Thursday	Friday
10-12	Data Structures	Data Structures (Lecture)		Data Structures (Lecture)	
12-1		Forest Walk		break	tutoring
1-3	Comp. Problem Solving (Evans 2617)	Data Structures (Lab, Practice)		Data Structures (Lab, Practice)	

Textbooks

Required

- First-time DSA students - zyBooks: Data Structures & Algorithms in C++
- Repeat DSA students The Algorithm Design Manual, 3rd edition, by Steven Skiena
- Cracking the Coding Interview – 189 programming questions and solutions, Gayle Laakmann McDowell, 6th edition (CCI)
  ISBN-13: 978-0984782857

Optional

- Grokking Algorithms by Aditya Barghava

List of Topics

Concrete Examples	Synthesis / Lateral Thinking	
Week 1	A Self-Guided Tour RAM Model, Lists Arrays, Links, Counters Interfaces, CRUD Operations	zyBooks and Git: practice and warmup Pair Programming: what roles? what rules?

Week 2	Implementation vs. Interfaces Stacks, Queues Key-Value Maps, Hashing Collisions, Links	Time-Space and Other Dualities, Resources How to trade off time for space, vice versa? Combining two or more data structures Counting and visualizing resources
Week 3	Asymptotic Analysis (Big O) Trees, Relations, Searching Traversals: Bread-first, Depth-First Heaps (Priority Queues)	Recursion and Iteration Cleaning Up: Invariants, Rebalancing Combining two or more data structures Overview of Final Project: Problem-Finding

Week 4	Rebalancing Trees Red-Black Trees Rebalancing Heaps Inventing New Data Structures	An Exotic Zoo: Fibonacci Forest, Skip Lists Amortized Analysis Combining two or more data structures Final Project Work: Collecting / Mocking Datasets
Week 5	Maximum Subset Sum Dynamic Programming Interval Scheduling Satisficing (Greedy)	Subproblems Memoizing / Caching Combining two or more algorithms Final Project Work: Collecting Datasets

Week 6	Graphs Topological Sort, Dependencies Shortest Path (Dijkstra's) Network Flow	MapQuest / Google Maps Find algorithms in the news Final Project Work: Designing Operations
Week 7	Sorting, with Heaps and Trees Merge Sort Quicksort Radix Sort	Randomized analysis Combining two sorts Joke Sorts: delay as much as possible Art: Pixel sorting

Week 8	Database Indexes, Search Page Rank Linear Systems of Equations Constraint Satisfaction	Boggle / Scrabble Final Project Work: Analysis and Correctness Find linear systems / constraints in real-life Connect DS & A to another discipline
Week 9	Karatsuba / Matrix Multiplication Divide and Conquer Streaming Algorithms Parallel Merge Sort	Data Privacy Find algorithms in the news Parallelize a previous algorithm Final Project Work: Analysis and Correctness

Week 10	Bonus Topics / Catchup Week Quantum DS & A Recommender Systems Collaborative Filtering	Finding Real World Examples: Go Local, Go Deep Final Project Work Synthesizing notes

Summary

You will have the freedom and responsibility to shape your own learning.

Computer technology has an impact on almost anything we do, and data structures and algorithms are central to advanced study in computer science and to building large complex systems. When applying for positions such as Software Developer or Software Engineer, you will probably find that this program is key to solving the problems of medium to large sized organizations. It is also the most commonly asked material in technical job interviews. In this program, you'll learn about ways to organize data (we'll see various data structures such as Lists, Trees, Graphs), ways to compare algorithms (we'll focus on space and time complexity using big O) and efficiently solve programming problems (sequential algorithms and parallel algorithms). It will also give you a chance to practice software design skills, both in the object-oriented and functional paradigms, and programming implementation skills.

We are so excited to work with all of you in this awesome program.

Learning Goals (Data Structures)

- Be able to combine data structures and algorithms effectively

- Learn technical and communication skills and build appropriate confidence to prepare you for a software job
- Improve your programming skills in Java or a language of your choice
- Improve your software design skills.
- Be able to argue, either formally or informally, for correctness of heuristics.
- Work collaboratively in teams, including pair programming

Learning Outcomes

By the conclusion of this course, each student should be able to:

- Describe the basic concepts of data structure and algorithms
- Conduct, critique, and submit conversations through AI chat to build up and verify their understanding.
- Co-create a class discussion by asking and answering questions of their classmates; building a class knowledge graph.
- Use IntelliJ for Java development (or another IDE for other languages) with autocomplete
- Commit and push to a git repository to develop software, collaborate with teams, and submit assignments.
- Find and discuss counterexamples and test cases that would exercise the "edges" of data structures and algorithms.
- Describe and demonstrate the techniques of asymptotic analysis to measure time, space, and other resources.
- Analyze the performance of searching, sorting, and any algorithm
- Implement searching algorithms such as: sequential searching and binary searching
- Describe the RAM model of computation, arrays and linked nodes.
- Implement data structures including: linked lists, trees, graphs, hash maps, sets, stacks, queues, heaps
- Differentiate an abstract data type (ADT) interface operations, such as create, read, update, and delete
- Understand and apply generic (class / template) types.
- Identify and "reverse engineer" the data structures and algorithms that might be needed by organizations, businesses, or websites that you encounter regularly and that improve community life.
- Implement several sorting algorithms such as: bubble sort, selection sort, insertion sort, merge sort, heap sort, and quick sort
- Compare and contrast different data structures.

- Design or adapt data structures and algorithms for specialized, real-world problems
- Discuss various social and ethical issues related to software development
- Collaborate with other students to complete as a final project a moderate-sized proof-of-concept incorporating four or more data structures and algorithms.

Policies

COVID

Health and Well-Being are central to student success, and at Evergreen we are committed to creating and maintaining a learning and working environment that is healthy, accessible, and equitable.

Any program/class member who feels ill with headache, excessive fatigue, fever, coughing, congestion, or other symptoms should stay home, and notify the faculty.

If you have questions or concerns please contact your faculty. If you wish to report a covid safety issue contact covid@evergreen.edu

Statement of Inclusion

We welcome everyone regardless of race, gender, abilities, prior experience, language fluency, or other areas of diversity. We will all promote a cooperative and supportive atmosphere within our program that helps to ensure that all people have opportunity and encouragement to speak freely. We will treat each person with civility and respect, especially when disagreeing with someone's ideas, attitudes or assumptions. We will remain sensitive to issues of racism, classism, sexism, abilism, homophobia, transphobia, and other forms of discrimination within the program context, and it is ok to make mistakes.

In an Evergreen "learning community" we are all co-learners present in good will with the opportunity to learn from each other, and we all have a responsibility to cultivate conditions for collaborative learning. Community is a state of being in the world that

requires our attention and work to achieve. In a learning community, we collaborate to expand our knowledge and abilities. Collaborative learning develops a set of skills and experiences that will equip you to deal with a broad array of situations in future learning and work of almost any kind.

Inclusive Practices Policy

Generous response to others' requests for inclusive practices counts people in. Requests that someone speak more loudly or slowly, read aloud, follow inclusivity guidelines for presentations, write in large letters on the board, or enable captioning when available are requests to be included.

By minimizing the presence of allergens and sensitivity-triggering fragrances on campus, Evergreen students, staff, and faculty make shared spaces available to people who live with otherwise-disabling conditions. To that end, please don't wear strongly-scented products or smoke-saturated clothing in classrooms and offices. Read Evergreen's Indoor Air Quality Policy (Air quality policy http://collab.evergreen.edu/policies/policy/airquality).

College-wide policies allow smoking (including e-cigarettes) only in designated campus areas. Read Evergreen's Smoking Policy (Smoking information https://evergreen.edu/facilities/smokers-information-center).

Summary

You will have the freedom and responsibility to shape your own learning.

Computer technology has an impact on almost anything we do, and data structures and algorithms are central to advanced study in computer science and to building large complex systems. When applying for positions such as Software Developer or Software Engineer, you will probably find that this program is key to solving the problems of medium to large sized organizations. It is also the most commonly asked material in

technical job interviews. In this program, you'll learn about ways to organize data (we'll see various data structures such as Lists, Trees, Graphs), ways to compare algorithms (we'll focus on space and time complexity using big O) and efficiently solve programming problems (sequential algorithms and parallel algorithms). It will also give you a chance to practice software design skills, both in the object-oriented and functional paradigms, and programming implementation skills.

We are so excited to work with all of you in this awesome program. Learning Goals (Data Structures)

```
Be able to combine data structures and algorithms effectively
Learn technical and communication skills and build appropriate confidence to prepare you for a software job
Improve your programming skills in Java or a language of your choice
Improve your software design skills.
Be able to argue, either formally or informally, for correctness of heuristics.
Work collaboratively in teams, including pair programming
```

Learning Outcomes

By the conclusion of this course, each student should be able to:

```
Describe the basic concepts of data structure and algorithms
Conduct, critique, and submit conversations through AI chat to build up and verify their understanding.
Co-create a class discussion by asking and answering questions of their classmates; building a class knowledge graph.
Use IntelliJ for Java development (or another IDE for other languages) with autocomplete
Commit and push to a git repository to develop software, collaborate with teams, and submit assignments.
Find and discuss counterexamples and test cases that would exercise the "edges" of data structures and algorithms.
Describe and demonstrate the techniques of asymptotic analysis to measure time, space, and other resources.
Analyze the performance of searching, sorting, and any algorithm
Implement searching algorithms such as: sequential searching and binary searching
Describe the RAM model of computation, arrays and linked nodes.
Implement data structures including: linked lists, trees, graphs, hash maps, sets, stacks, queues, heaps
Differentiate an abstract data type (ADT) interface operations, such as create, read, update, and delete
```

```
Understand and apply generic (class / template) types.
Identify and "reverse engineer" the data structures and algorithms that might be
needed by organizations, businesses, or websites that you encounter regularly and
that improve community life.
Implement several sorting algorithms such as: bubble sort, selection sort, insertion
sort, merge sort, heap sort, and quick sort
Compare and contrast different data structures.
Design or adapt data structures and algorithms for specialized, real-world problems
Discuss various social and ethical issues related to software development
Collaborate with other students to complete as a final project a moderate-sized
proof-of-concept incorporating four or more data structures and algorithms.
```

Policies COVID

Health and Well-Being are central to student success, and at Evergreen we are committed to creating and maintaining a learning and working environment that is healthy, accessible, and equitable.

Any program/class member who feels ill with headache, excessive fatigue, fever, coughing, congestion, or other symptoms should stay home, and notify the faculty.

If you have questions or concerns please contact your faculty. If you wish to report a covid safety issue contact covid@evergreen.edu Links to an external site. Statement of Inclusion

We welcome everyone regardless of race, gender, abilities, prior experience, language fluency, or other areas of diversity. We will all promote a cooperative and supportive atmosphere within our program that helps to ensure that all people have opportunity and encouragement to speak freely. We will treat each person with civility and respect, especially when disagreeing with someone's ideas, attitudes or assumptions. We will remain sensitive to issues of racism, classism, sexism, abilism, homophobia, transphobia, and other forms of discrimination within the program context, and it is ok to make mistakes. In an Evergreen "learning community" we are all co-learners present in good will with the opportunity to learn from each other, and we all have a responsibility to cultivate conditions for collaborative learning. Community is a state of being in the world that requires our attention and work to achieve. In a learning community, we

collaborate to expand our knowledge and abilities. Collaborative learning develops a set of skills and experiences that will equip you to deal with a broad array of situations in future learning and work of almost any kind. Inclusive Practices Policy

Generous response to others' requests for inclusive practices counts people in. Requests that someone speak more loudly or slowly, read aloud, follow inclusivity guidelines for presentations, write in large letters on the board, or enable captioning when available are requests to be included.

By minimizing the presence of allergens and sensitivity-triggering fragrances on campus, Evergreen students, staff, and faculty make shared spaces available to people who live with otherwise-disabling conditions. To that end, please don't wear strongly-scented products or smoke-saturated clothing in classrooms and offices. Read Evergreen's Indoor Air Quality Policy (Air quality policy http://collab.evergreen.edu/policies/policy/airquality).

College-wide policies allow smoking (including e-cigarettes) only in designated campus areas. Read Evergreen's Smoking Policy (Smoking information https://evergreen.edu/facilities/smokers-information-center).