

Proposal

Data Structures

Andrew Rosen

1. My aim is to replace the current textbook used in CIS 2168 (Data Structures) with a free resource the students can utilize. Data Structures is one of the fundamental topics in Computer Science and is taught in some form at *every* school that offers a Computer Science degree. Furthermore, it is one of the two courses (the other being the Algorithms course) around which large tech companies tailor the content of technical interviews. Students are stretched for cash as it is and additional costs of textbooks can serve as both a real and perceived barrier to entry. Additionally, I want to host this project on GitHub so that other instructors can create their own version of the textbook and make additional contributions to the text.
2. You can view my progress on the textbook by clicking this sentence. You can find the current table of contents attached.
3. The target audience at Temple is all students required to take CIS 2168, which is every student enrolled in the CIS or IST major or minor program. This is about 175 students a semester. This course is required in some fashion at all universities that offer a computer science degree. Data Structures is sometimes referred to as CS2, where CS1 would be the intro to programming course. There is some variation among universities as to which course gets which specific topic, but that is more a matter of timing than anything else. This is a roundabout way of saying that this book is applicable to any student getting a CS or CS related degree who has taken their introductory programming courses.
4. We currently use “Data Structures: Abstraction and Design Using Java” by Koffman and Wolfgang, which costs \$125. Other textbooks for this course exist, such as Prichard and Carrano’s “Data Abstraction and Problem Solving with Java: Walls and Mirrors” for a comparable price. As mentioned above, every university that grants CS degrees teaches the topics in this course. It is a required part of the CS curriculum and is offered every semester at every university I’ve been to.
5. Dr. Amy J. Ko keeps a list of Professors who do research in Computer Science education (Computing Education Research) many of these people would be good potential reviewers. Other, more specific people I can reach out to:
 - (a) Cliff Shafer
 - (b) Tammy Pirmann

- (c) Brianna Morrison
- (d) Austin Cory Bart
- (e) Melinda McDaniel

Contents

1	Introduction	7
1.1	What is a Data Structures Course	7
1.2	Why This Book?	7
1.2.1	Where Does This Book Fit Into a Computer Science Curriculum	7
1.2.2	What Are My Base Assumptions about the Reader?	7
1.3	To The Instructor	8
1.4	To The Student	8
2	The Array	9
2.1	Array Operations	9
2.2	Finding Values in an Array	9
3	Analyzing Algorithms	11
3.0.1	Cost	11
3.1	Big O Notation	11
3.1.1	Space Complexity	11
3.2	The Formal Mathematics of Big O Notation	11
3.3	Other Notations	11
4	Lists	13
4.1	What is a list?	13
4.2	ArrayLists	14
4.2.1	Generics	14
4.2.2	Building an ArrayList	14
4.2.3	More Restrictive or Permissive Generics	14
4.3	LinkedLists	14
4.3.1	Building a LinkedList	14
4.4	Analysis	14
5	Stacks	15
5.1	Building a Stack	15
5.2	Mazes - Stacks and Backtracking	15
5.3	Discrete Finite Automata	15
6	Queues	17
6.1	Circular Arrays	17

7	Recursion	19
7.1	Recursive Mathematics	19
7.2	Recursive Problem Solving	19
7.2.1	Recursive Backtracking	19
7.2.2	Recursive Combinations	19
7.3	Recursion and Puzzles	19
7.4	Recursion and Art	19
7.5	Recursion and Nature	19
8	Trees	21
8.1	Binary Search Trees	21
8.2	Heaps	21
8.2.1	Priority Queues	21
8.3	Trees and Heaps in Java	21
9	Sorting	23
9.1	Quadratic-Time Algorithms	23
9.1.1	Bubble Sort	23
9.1.2	Selection Sort	23
9.1.3	Insertion Sort	23
9.2	Log-Linear Sorting Algorithms	23
9.2.1	Tree Sort	23
9.2.2	Heap Sort	23
9.2.3	Quick Sort	23
9.2.4	Merge Sort	23
9.3	Unique Sorting Algorithms	23
9.3.1	Shell Sort	23
9.3.2	Radix Sort	23
9.4	State of the Art Sorting Algorithms	23
9.4.1	Tim Sort	23
9.4.2	Quick Sort	23
9.4.3	Distributing and Parallelization	23
10	Sets and Maps	25
10.1	Sets	25
10.2	Maps	25
10.3	Hash Tables	25
10.3.1	Creating a Hash Function	25
10.4	Map Reduce	25
11	Graphs	27
11.1	Introduction and History	28
11.2	Qualities of a Graph	28
11.2.1	Undirected Edges	28
11.2.2	Directed Edges	28
11.2.3	Weighted Edges	28
11.3	Directed Acyclic Graphs	28
11.4	Building a Graph	28
11.4.1	Adjacency List	28
11.4.2	Adjacency Matrix	28

11.5	Graph Algorithms	28
11.5.1	Searching and Traversing	28
11.5.2	Shortest Path	28
11.5.3	Topological Sorting	28
11.5.4	Minimum Spanning Trees	28
11.6	Graphs, Humans, and Networks	28
11.6.1	The Small World	28
11.6.2	Scale Free Graphs	28
11.7	Graphs in Art and Nature - Voronoi Tessellation	28
11.8	Distributed Hash Tables	28
11.9	A Nontechnical Introduction to NP-Completeness	28
11.9.1	The Traveling Salesperson Problem (TSP)	28
11.9.2	The Longest Path Problem	28
11.9.3	The Rudrata/Hamiltonian Path Problem	28
12	Other Data Structures	31
12.1	Skip Lists	31