

Data Structures

Andrew Rosen

Contents

1	Introduction	5
1.1	To The Instructor	5
2	Lists	7
2.1	What is a list?	7
2.2	ArrayLists	8
2.2.1	Building an ArrayList	8
2.3	Big O	8
2.3.1	Cost	8
2.3.2	Space Complexity	8
2.3.3	Formal Mathematics	8
2.4	LinkedLists	8
2.4.1	Building a LinkedList	8
2.5	Analysis	8
3	Stacks	9
3.1	Building a Stack	9
3.2	Mazes	9
4	Queues	11
4.1	Discrete Finite Automata	11
5	Recursion	13
5.1	Recursive Mathematics	13
5.2	Recursive Problem Solving	13
5.2.1	Recursive Backtracking	13
5.2.2	Recursive Combinations	13
6	Trees	15
7	Sorting	17
7.1	Quadratic-Time Algorithms	17
7.2	Insertion Sort	17
7.3	Bubble Sort	17
7.4	Recursive Sorting Algorithms	17
7.5	Unique Sorting Algorithms	17
7.5.1	Shell Sort	17
7.5.2	Radix Sort	17
7.6	State of the Art Sorting Algorithms	17

8	Sets and Maps	19
9	Graphs	21
10	Other Data Structures	23

Chapter 1

Introduction

Education in Computer Science is based around three core topics: how to solve problems, how to organize data the for solving problems, and how to formulate solutions for problems in a way that a computer can understand.

Data Structures is all about defining the different ways we can organize data.

1.1 To The Instructor

Chapter 2

Lists

The first data structure we will be studying is the list. The list is by far the most relateable data structure, as humans deal with lists on a regular basis

2.1 What is a list?

When you get right down to it, lists are defined by order.

```
public static <E> boolean isPermutation(List<E> listA, List<E> listB) {  
  
    if(listA.size() != listB.size()) {  
        return false;  
    }  
    for(int i = 0; i < listA.size() ; i++){  
        E item = listA.get(i);  
        int countA = 0;  
        int countB = 0;  
  
        for (E element : listA) {  
            if(item.equals(element)){  
                countA++;  
            }  
        }  
        for (E element : listB) {  
            if(item.equals(element)){  
                countB++;  
            }  
        }  
        if(countA != countB) {  
            return false;  
        }  
    }  
    return true;  
}
```

2.2 ArrayLists

2.2.1 Building an ArrayList

2.3 Big O

2.3.1 Cost

2.3.2 Space Complexity

2.3.3 Formal Mathematics

2.4 LinkedLists

2.4.1 Building a LinkedList

2.5 Analysis

Chapter 3

Stacks

3.1 Building a Stack

3.2 Mazes

Chapter 4

Queues

4.1 Discrete Finite Automata

Chapter 5

Recursion

5.1 Recursive Mathematics

5.2 Recursive Problem Solving

5.2.1 Recursive Backtracking

5.2.2 Recursive Combinations

Chapter 6

Trees

Chapter 7

Sorting

7.1 Quadratic-Time Algorithms

7.2 Insertion Sort

7.3 Bubble Sort

7.4 Recursive Sorting Algorithms

7.5 Unique Sorting Algorithms

7.5.1 Shell Sort

7.5.2 Radix Sort

7.6 State of the Art Sorting Algorithms

Chapter 8

Sets and Maps

Chapter 9

Graphs

Chapter 10

Other Data Structures