

Introduction to Data Structures

1 An Introduction to Data Structures

Data Structures are in the most simple to understand definition, a method of organizing data. Basic Data Structures that we have seen already include arrays and the various "Deque"s¹ and IntLists. These data structures are perfectly fine; however, we may want to store our data in a different manner that would be more optimized. In this chapter, we will go through the data structures, their runtimes, and pros and cons of using certain data structures.

2 Introductory Data Structures

Before we discuss more complicated data structures, we will discuss more basic ones, specifically **LinkedLists**, **Arrays**, **Stacks**, and **Queues**.

LinkedLists were one of the first data structures we went over, in the form of IntLists. A LinkedList is basically a set of nodes, each with a next and previous pointer. With all the full optimizations, a LinkedList takes $O(N)$ time to find an item, $O(1)$ to insert, and $O(N)$ to delete an item. LinkedLists, along with arrays, are the data structures that most other data structures are derived from.

Arrays are "containers" that can hold a fixed number of objects- you establish how many items it can hold when instantiating it. Arrays have a useful counterpart, ArrayLists, which are arrays that resize themselves automatically. An arraylist takes $O(1)$ time to find an item (given an index), $O(N)$ to find a specific item (not given the index), $O(N)$ to insert into an array (because of resizing), and $O(N)$ to delete because you would have to shift over elements (given the index) and find the element (not given the index).

Stacks are First in Last Out Data Structures (FILO). This means that if you insert item A into a stack followed by item B and remove all the items, B will come out before A. The *push* operation of a stack puts an item on the top, the *peek* operations returns the most recently put item, and the *pop* operation removes the first item. All of these operations take $O(1)$ time.

The final basic data structure is a Queue. A queue is a First in First Out Data Structure (FIFO). This means that if you insert item A in a queue followed by item B and remove all the items, item A will come out before item B.

¹The Deque's are from UC Berkeley's Computer Science 61B course taught under Josh Hug, it was not discussed explicitly in this material, the basic concept is similar to that of a LinkedList (which we will reference soon).