

Data Structures & Algorithms - A Review

Kevin Chung

December 2016

Contents

1	Preface	1
2	Running Time	1
2.1	Example: Printing an Array	1
2.2	Example: Bubble Sort	1
3	Data Structures	2
3.1	Lists	2
3.2	Stacks	2
3.3	Queues	2

1 Preface

This review package/compilation/set or whatever you wish to call it is a way for me, Kevin to review for my coming semesters of computing science courses. These are the fundamentals of which I have learned in my past couple years of undergraduate education. There may be errors, I'm not perfect but I will try my best to convey the most accurate information. Most of the information will be comprised from my personal notes or from a professor's notes. If I find that information is lacking on a topic or I don't understand it enough, the Internet may prove to be useful.

2 Running Time

One of the earliest concepts programmers learn is running time specifically in regards to algorithms and comparing algorithms. Although there are many types of notations to describe run time, I will be using Big O notation as this describes for lack of a better time, the worst case scenario or the upper bound on the growth rate of a function. For a very primitive example, lets iterate through a whole array of numbers and print them out.

2.1 Example: Printing an Array

```
int prntArr(int Arr[], int n){
    for(int i=0; i<n; i++){
        std::cout<<Arr[i]<<std::endl;
    }
}
```

The above code simply iterates through the array via the for loop using the indices from 0 to n (the size of the array). As a result, if we had an array of size n, and arbitrary number this time, we would have to print out n numbers. Thus, the upper bound of this algorithm would be $O(n)$ or Big-O of n as there can be no greater number of outputs larger than n. In another way of analyzing it, we only call the cout function n times at most. The O in Big-O stands for Order; we will see another example of it in the next example.

2.2 Example: Bubble Sort

Personally, when I started my first Computing science course, I wasn't thinking about optimization. I was focused on getting the right answer and always amazed at the results. I remember very clearly how I would use bubble sort (I find this pretty funny after years of programming) to sort a list of numbers. In this example, I will analyze bubblesort().

```

void bubblesort(int list[], int n){
    for(int i=0; i<n-1; i++){
        for(int j=0; j<n-1-i; j++){
            if(list[j]>list[j+1]){
                int tempval = list[j];
                list[j] = list[j+1];
                list[j+1] = tempval;
            }
        }
    }
}

```

In the above code, we have parameters `list[]` and an `int n` which are respectively the list we wish to sort and the size `n`.

3 Data Structures

3.1 Lists

3.2 Stacks

3.3 Queues