

# Homework 6

GWU CSCI 1112 - Spring 2019

November 8, 2019

## 1 Introduction

This homework is intended to reinforce your understanding of Big-O and run-times for algorithms. It is comprised of various different exercises that will ask you to find the run-times of loops, explain the run-times of algorithms you have seen in class, and improve algorithms.

### 1.1 Deadline

November 15, 2019 at 11:59pm

### 1.2 Submission

You must turn in a pdf file with your answers on it.

### 1.3 Grading Rubric

Grading will be based off of both correct answers as well as answer completion. Be sure to show all over your work and explain things thoroughly. Incomplete answers will not receive full credit.

### 1.4 Plagiarism

We will use a set of automated tools specifically designed to analyze solutions for plagiarism. If you copy code from another source, classmate or website, there is a very high probability that these tools will flag your work as plagiarized. You are permitted to discuss the problems at a high level; however, you must write your own solution. If you do not share answers or outright borrow answers from a website, you will have no problem with the plagiarism filter.

## 2 Run-time and Big-O Exercises

### 2.1 Calculate the Big-O run time of these loops

#### 2.1.1

```
i=0;
While i < n {
  For (j=0;j<n;j++){
    Print(this is a loop)
  }
  i = i*2;
}
```

#### 2.1.2

```
Array x = new array[n]
Count = 0;
For (i=0;i<n;i++){
  Array[Count] = I;
  Count ++
  If ( i%3 =0) {
    While (Count !=0) {
      Count --
      Array[Count] = 0
    }
  }
}
```

### 2.2 Algorithm run-time (Big-O) questions

#### 2.2.1

What is the run-time of bubble sort? Explain why in detail.

### 2.2.2

What is the run-time of running binary search on a sorted array? Explain why in detail.

### 2.2.3

What is the Worst case run-time of Quicksort? Why? What is the average case run-time of Quicksort? Why?

## 2.3 Fibonacci Sequence

### 2.3.1

Find the Big-O run-time of this program

```
Public int Fibb ( int x ){  
    If(x==1) return 1;  
    If (x==0) return 0;  
    Return Fibb(x-1)+Fibb(x-2);  
}
```

### **2.3.2**

Rewrite this program so that it has a much smaller run-time.

Hint: use an array to store information.  $O(n)$  is the optimal solution

### **2.3.3**

Explain the run time of your program.