Name: Puri Sangnawakit ID: 682115036

Tutorial 3 Algorithm Efficiency and Midterm review (10 points)

1. What is the big-O of the following snippet (4 points)

1.1

```
int result = 0
    int i = 1
while i < n
     if n % i == 0
        result += i
     end
i += 1
end
return result
```

O(n)

1.2

```
if array[0] == null
  return true
else
  return false
end
```

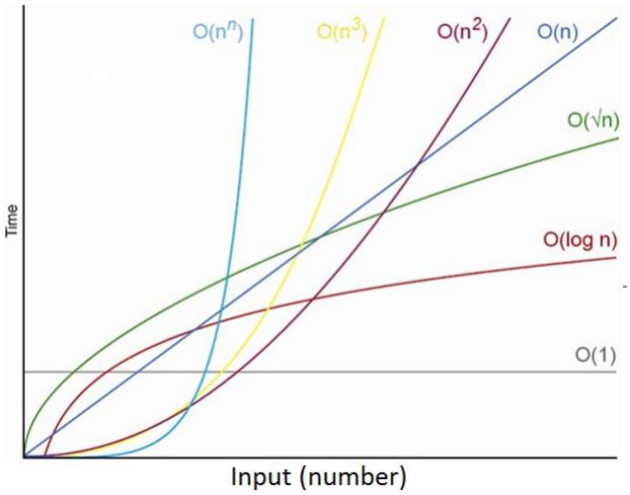
O(1)

1.3

```
public static int doSomething(int[] arr, int x){

    int size = arr.length;
    for(int i=0;i<size;i++){
       if(arr[i] ==x){
          return i;
       }
    }
    return -1;
 }
```

O(n)



1.4 According to above comparison figure, which function represents the fastest algorithm?

O(1)

2. Write a Java program that read a data file (you can download from the link here (6points) https://www.dropbox.com/s/chnpp0kkvpbbyfb/data.txt?dl=0

Your program must have a method call "mySearch" which responses to find for all the value in the given data file that are greater than 0.5. Below is an example output of the program from a different data file.

```
>Total number of values read: 15103
>Number of value > 0.5 is: 1343
```

What is the Big O of your method mySearch? **O(n)**

Copy and paste your java source code here

```
import java.util.*;
import java.io.*;

public class Search {
    public static void main(String[] args) throws IOException{

        // 1. Reading data from a file
        String filename = args[0];
        File file = new File(filename);

        Scanner myScan = new Scanner(file);
        myScan.useDelimiter("[,\\s]+");

        // 2. ArrayList to store the values
        ArrayList<Double> dataList = new ArrayList<>();

        while(myScan.hasNextDouble()){
            double value = myScan.nextDouble();
            dataList.add(value);
        }
        myScan.close();

        int result = mySearch(dataList);

        System.out.println("Total number of values read: " +
dataList.size());
        System.out.println("Number of values > 0.5: " + result);
    }

    public static int mySearch(ArrayList<Double> values) {
        // Number of value that > 0.5
        int countValue = 0;
        for (double val : values){
            if(val > 0.5){
                countValue++;
            }
        }
        return countValue;
    }
}
```