Punto 1

-ArrayMax:

```java
import java.util.Random;
public class Array
{
    public static void main(String [] args) {
       for(int i = 50; i<=70; i++) {
           long startTime = System.currentTimeMillis();
           ArrayMax(f(i));
           long estimatedTime = System.currentTimeMillis() - startTime;
           System.out.println(i+ " " + estimatedTime);
        }
    }
    public static void ArrayMax(int[]nums){
        recorrer(nums,0,nums[0]);
    }
    private static int recorrer(int[]nums, int i,int max){
        if (i==nums.length){
            //System.out.println(max);
            return max;
        }
        if(nums[i]>max){
            max=nums[i];
        }
        return recorrer(nums,i+1,max);
        }
        public static int[] f(int size) {
       int max = 5000;
       int[] array = new int[size];
       Random generator = new Random();
       for(int i = 0; i<size; i++) {
         array[i] = generator.nextInt(max);
        }
       return array;
    }
    }
```

-GroupSum:

```java
{
public static void main(String []args) {
      for(int i = 15; i<=35; i++) {
      long startTime = System.currentTimeMillis();
      groupSum(0, f(i), 10000);
      long estimatedTime = System.currentTimeMillis() - startTime;
      System.out.println(i + " " + estimatedTime);
   }
}
public static boolean groupSum(int start, int[] nums, int target) {
   if(start>=nums.length) {
       return target == 0;
   } else {
     return groupSum(start+1, nums, target  - nums[start])
     || groupSum(start+1, nums, target);
   }
 }
public static int[] f(int size) {
    int max = 5000;
    int[] array = new int[size];
    Random generator = new Random();
    for(int i = 0; i<size; i++) {
        array[i] = generator.nextInt(max);
       }
    return array;
   }
}
```

Fibonacci:

```
1
2  public class Fibonacci
3  {
4      public static void main(String [] args) {
5          for(int i = 28; i<=48; i++) {
6              long startTime = System.currentTimeMillis();
7              fibonacci(i);
8              long estimatedTime = System.currentTimeMillis() - startTime;
9              System.out.println(i+ " " + estimatedTime);
10         }
11     }
12     public static int fibonacci(int n) {
13         if(n<=1) {
14             return n;
15         } else {
16             return fibonacci(n-1) + fibonacci(n-2);
17         }
18     }
19 }
20
```

Punto 2

ArrayMax:
El tamaño del problema siempre va a ser constante.

GroupSum:
En este algoritmo, al igual que el anterior, lo que define la magnitud del problema es el tamaño del arreglo.

Fibonacci:
La variable n es la que va a definir el tamaño del problema en este algoritmo.

Punto 3

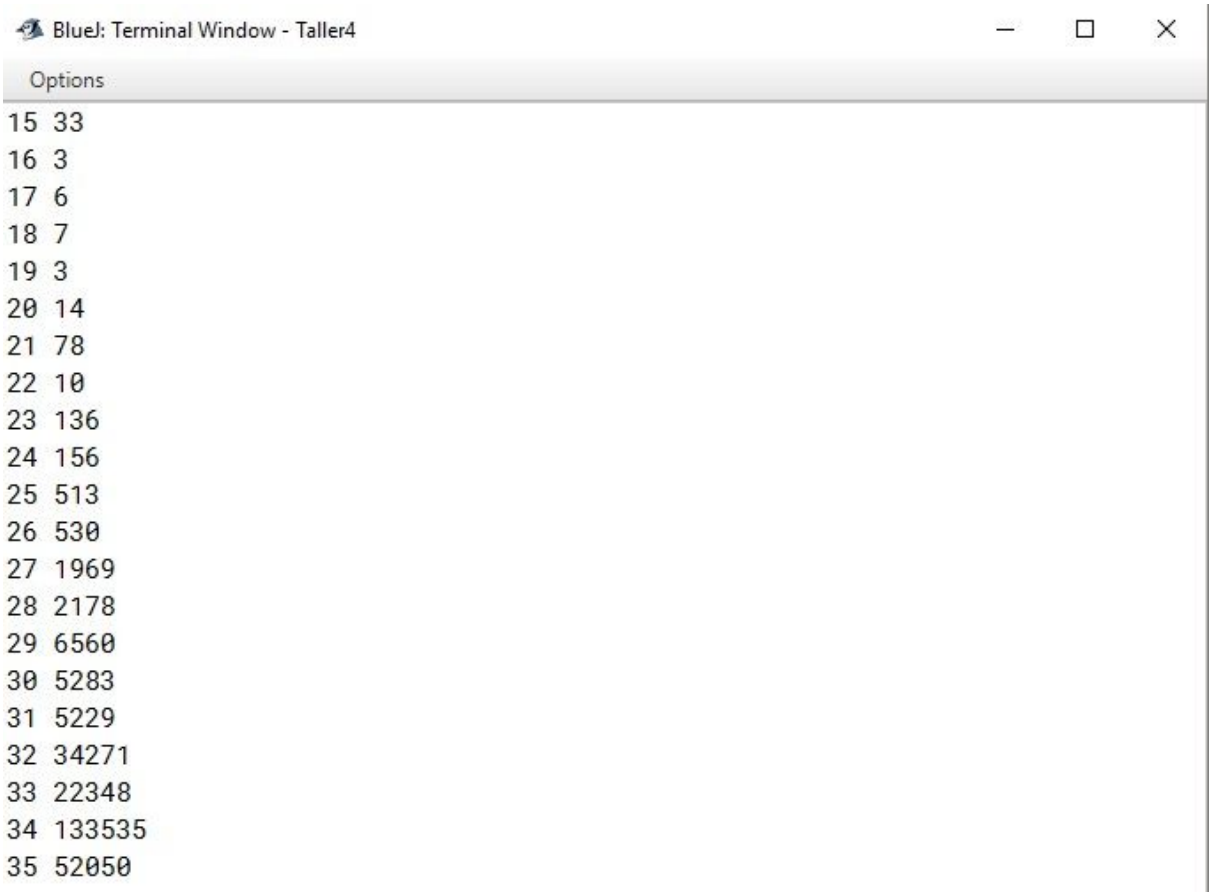
ArrayMax:Se probó con varios rangos y se llegó a la conclusión, dadas las pruebas, que el tiempo oscila entre 0 y 1 segundos siempre, por ende se consideró constante. Al final decidimos evaluar entre 50 y 70 para ejemplificar lo antes mencionado.

GroupSum:
Valores entre 15 y 35

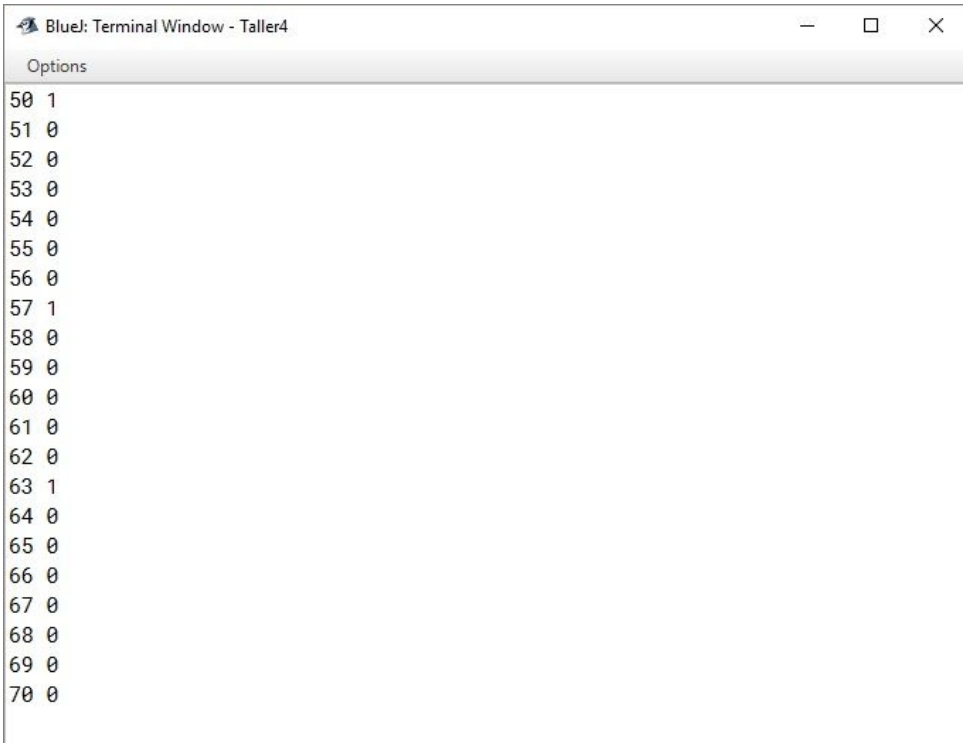
Fibonacci:
Valores entre 28 y 48

Punto 4

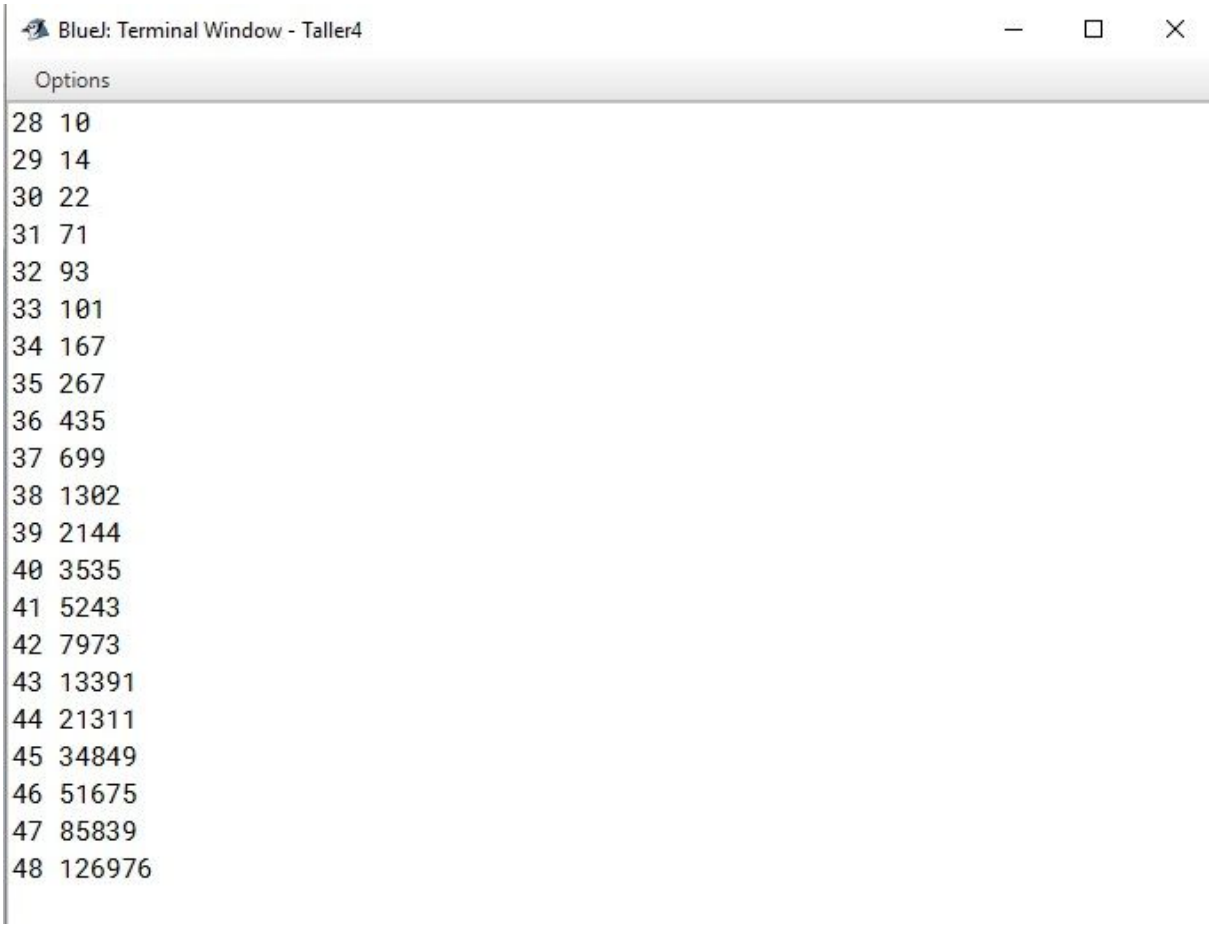
groupSum:



ArrayMax:

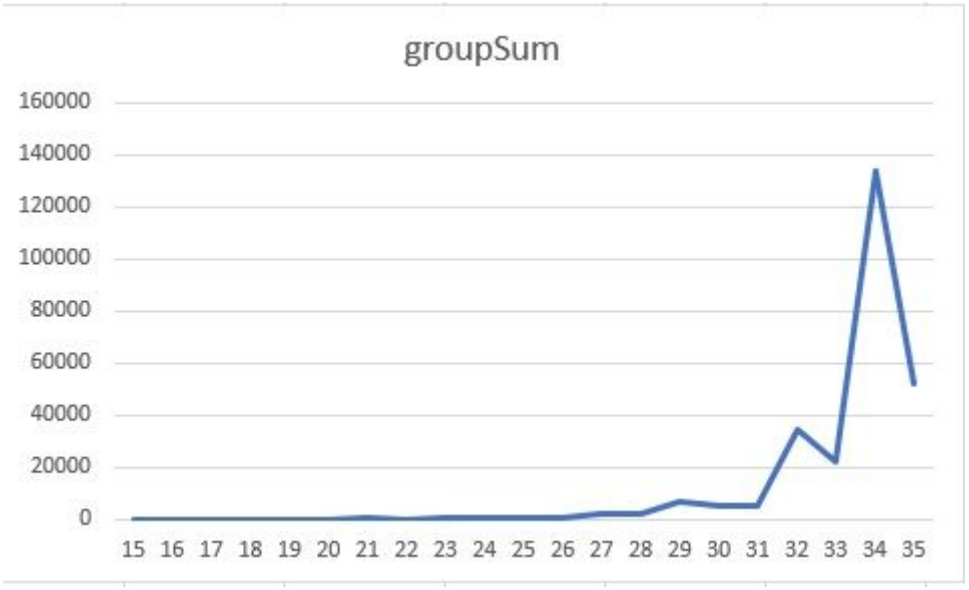


Fibonacci:

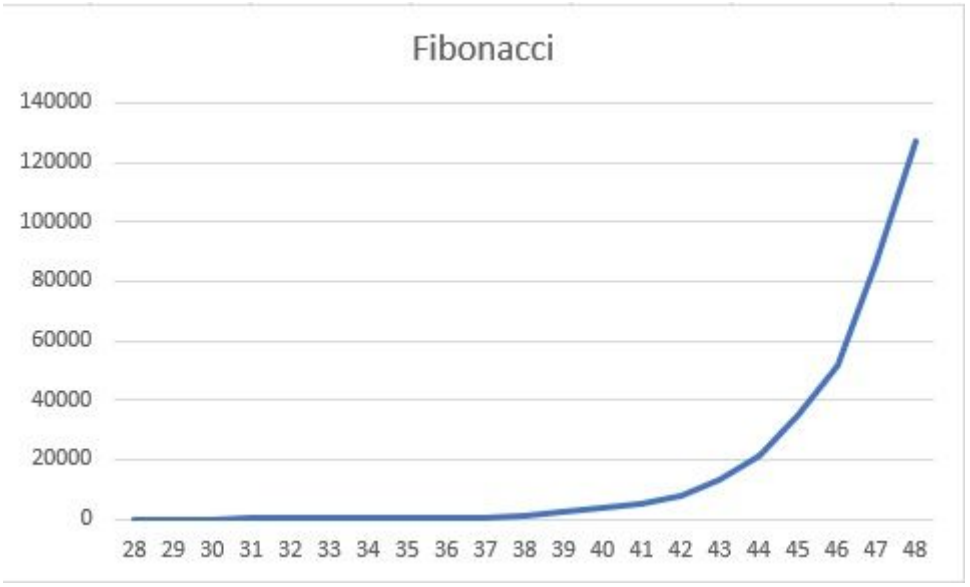


Punto 5

-groupSum:



-Fibonacci:



-ArrayMax: