

Punto 1

-ArrayMax:

```
1 import java.util.Random;
2 public class Array
3 {
4     public static void main(String [] args) {
5         for(int i = 50; i<=70; i++) {
6             long startTime = System.currentTimeMillis();
7             ArrayMax(f(i));
8             long estimatedTime = System.currentTimeMillis() - startTime;
9             System.out.println(i+ " " + estimatedTime);
10        }
11    }
12    public static void ArrayMax(int[] nums){
13        recorre(nums,0,nums[0]);
14    }
15    private static int recorre(int[] nums, int i,int max){
16        if (i==nums.length){
17            //System.out.println(max);
18            return max;
19        }
20        if (nums[i]>max){
21            max=nums[i];
22        }
23        return recorre(nums,i+1,max);
24    }
25    public static int[] f(int size) {
26        int max = 5000;
27        int[] array = new int[size];
28
29        Random generator = new Random();
30        for(int i = 0; i<size; i++) {
31            array[i] = generator.nextInt(max);
32        }
33        return array;
34    }
35 }
```

-GroupSum:

```
1 {
2     public static void main(String [] args) {
3         for(int i = 15; i<=35; i++) {
4             long startTime = System.currentTimeMillis();
5             groupSum(0, f(i), 10000);
6             long estimatedTime = System.currentTimeMillis() - startTime;
7             System.out.println(i + " " + estimatedTime);
8         }
9     }
10 }
11
12 public static boolean groupSum(int start, int[] nums, int target) {
13     if(start>=nums.length) {
14         return target == 0;
15     } else {
16         return groupSum(start+1, nums, target - nums[start])
17             || groupSum(start+1, nums, target);
18     }
19 }
20
21 public static int[] f(int size) {
22     int max = 5000;
23     int[] array = new int[size];
24     Random generator = new Random();
25     for(int i = 0; i<size; i++) {
26         array[i] = generator.nextInt(max);
27     }
28     return array;
29 }
```

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:

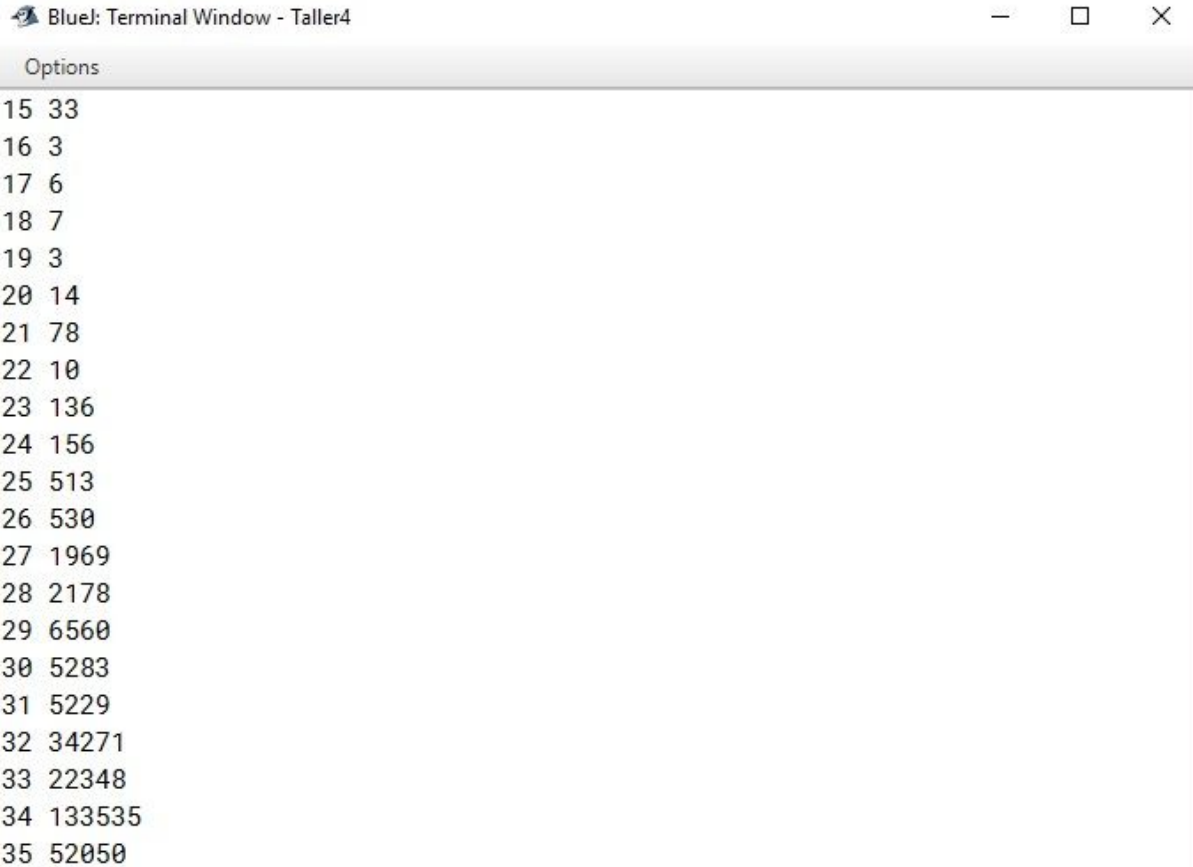
Valores entre 15 y 35

Fibonacci:

Valores entre 28 y 48

Punto 4

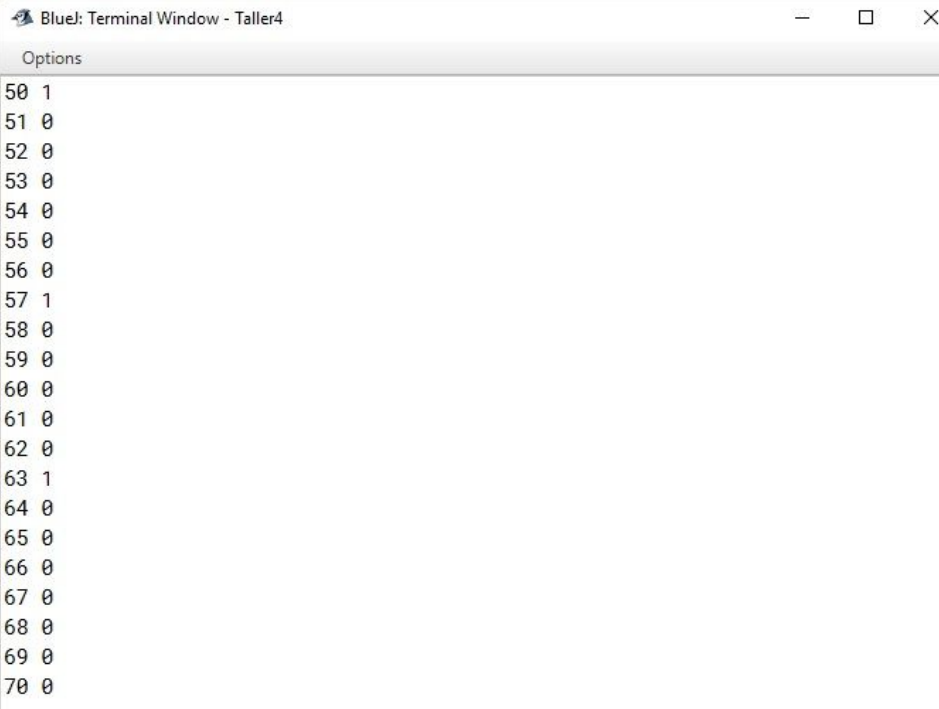
groupSum:



A terminal window titled "BlueJ: Terminal Window - Taller4" with standard window controls. Below the title bar is a tab labeled "Options". The terminal displays a list of 21 pairs of numbers, each on a new line, representing the output of the groupSum function for indices 15 through 35.

```
15 33
16 3
17 6
18 7
19 3
20 14
21 78
22 10
23 136
24 156
25 513
26 530
27 1969
28 2178
29 6560
30 5283
31 5229
32 34271
33 22348
34 133535
35 52050
```

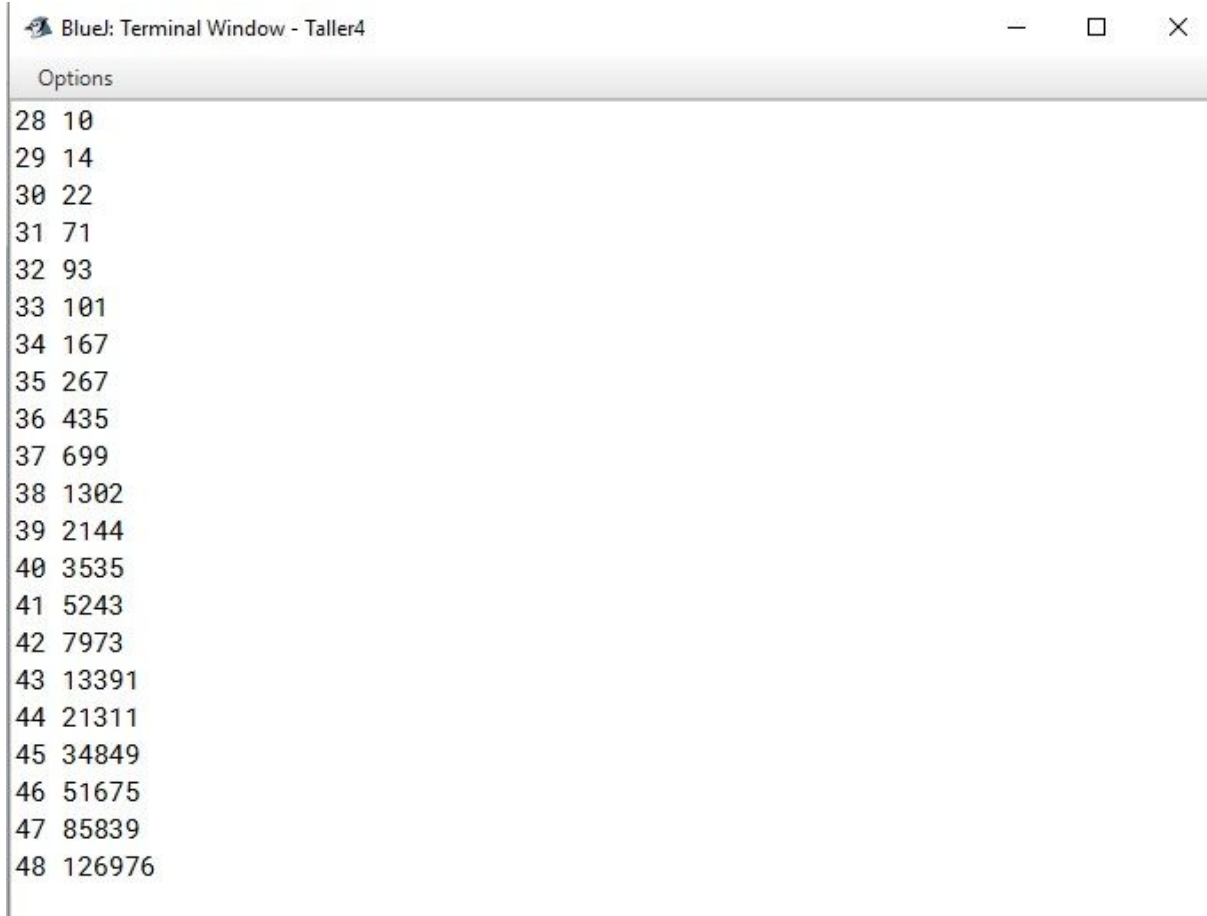
ArrayMax:



A terminal window titled "BlueJ: Terminal Window - Taller4" with standard window controls. Below the title bar is a tab labeled "Options". The terminal displays a list of 21 pairs of numbers, each on a new line, representing the output of the ArrayMax function for indices 50 through 70.

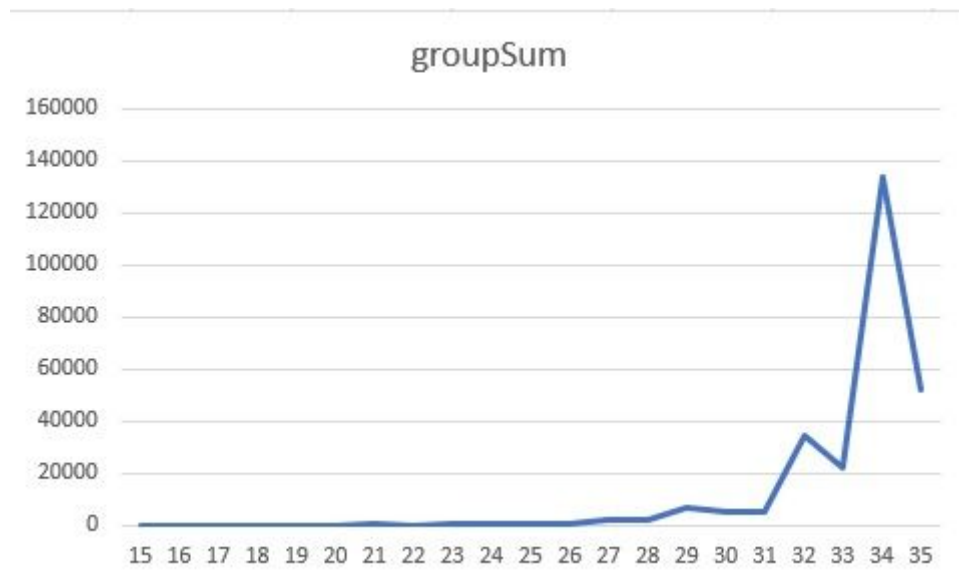
```
50 1
51 0
52 0
53 0
54 0
55 0
56 0
57 1
58 0
59 0
60 0
61 0
62 0
63 1
64 0
65 0
66 0
67 0
68 0
69 0
70 0
```

Fibonacci:

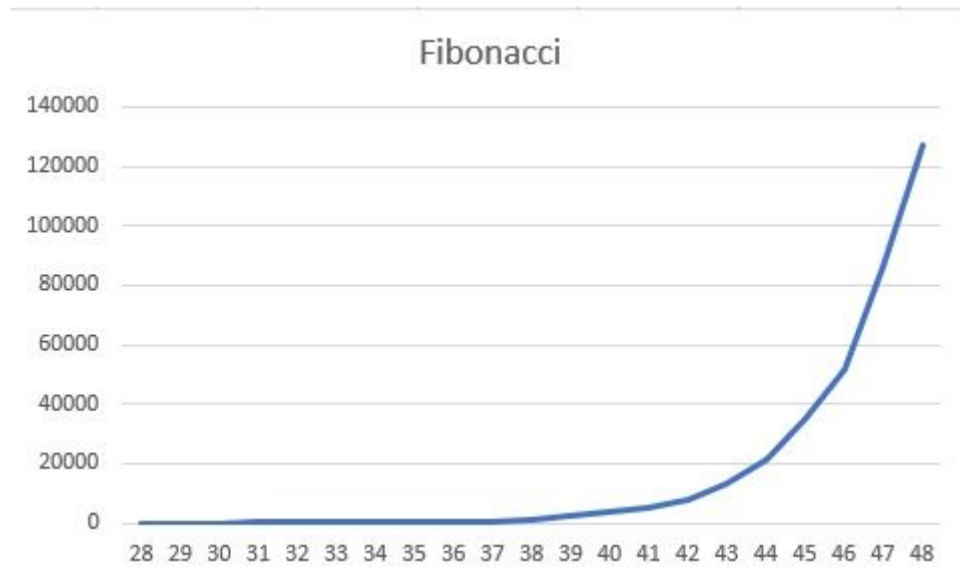


Punto 5

-groupSum:



-Fibonacci:



-ArrayMax:

