

Laboratorio 1

Felipe Álvarez Benítez
Andrés Ospina Patiño

Punto 2

RECURSION 1:

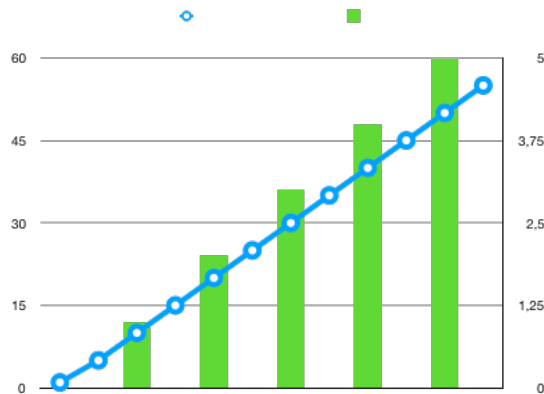
- Factorial: $T(n) = n * T(n-1)$
- bunnyEars: $T(n) = 2 + T(n-1)$
- Fibonacci: $T(n) = T(n-1) + T(n-2)$
- bunnyEars2: $T(n) = c + T(n-1)$
- triangle: $T(n) = c + T(n-1)$

RECURSION 2:

- groupSum: $T(n) = 2T(n-1)$
- groupSum6: $T(n) = 2T(n-1)$
- groupSumNoAdj: $T(n) = T(n-1) + T(n-2)$
- groupSum5: $T(n) = 2T(n-1)$
- splitArray: $T(n) = 2T(n-1)$

Punto 3.1 Calcular la complejidad asintotica para el ejercicio 1.2

R/ La complejidad asintotica del problema de como acomodar los rectangulos es de $O(2^n)$ y su ecuación de recurrencia es $T(n) = T(n-1) + T(n-2)$



Y si puede ser implementado en puerto antioquia en 2020 debido a que es un algoritmo

Punto 4.2

Pepito escribió el siguiente código usando recursión:

```
private int b(int[] a, int x, int low, int high) {
    if (low > high) return -1;
    int mid = (low + high)/2;
    if (a[mid] == x) return mid;
    else if (a[mid] < x)
        return b(a, x, mid+1, high);
    else
        return b(a, x, low, mid-1);
}
```

Respuesta a

4.3

```
1 int solucionar (int n, int a, int b, int c) {
2     if (n == 0 || (n < a && n < b && n < c))
3         return 0;
4     int res = solucionar( n-a, a, b, c ) + 1;
5     res = Math.max( res , solucionar(n-b, a, b, c)+1 );
6     res = Math.max( res , solucionar(n-c, a, b, c)+1 );
7     return res; }
```



```
1. public int formas(int n){
2. if(n <= 2) n;
3. return formas(n-1) +
4. formas(n-2);
```

4.5.2 ¿Cuántas instrucciones ejecuta el algoritmo en el peor de los casos?:

1. b. $T(n)=T(n-1)+T(n-2)+C$

4.6

```
1 public int suma(String n) { 2 return sumaAux(n, 0); 3}
4
```

```
5 private int sumaAux(String n, int i){
6   if (i >= n.length()) {
7     n.charAt(i) == n.charAt(i + 1)){
8     return __sumaAux(n,i+1)__;
9   }
10  }
11  }
12  return (n.charAt(i) - '0') + __sumaAux(n,i+1)__;
13 }
```

4.9 Considere el siguiente programa. ¿Cuál es la salida generada por $fun(11,5)$? Como un ejemplo: $fun(10,3)=20$.

Respuesta/ c. 22

4.10 Considere el siguiente programa. ¿Cuál es la salida para $fun(1,4)$? Como un ejemplo: $fun(1,2)=4$.

Respuesta / b. 6

4.11

```
1 int lucas(int n){
2   if(n == 0) return 2;
3   if(n == 1) return 1;
4   return lucas( n-1 ) + lucas(n-2);
5 }
```

Y su complejidad asintotica es $T(n)=T(n-1)+c$, que es $O(n)$

4.12