

Laboratorio Nro. 1

Escribir el tema del laboratorio

Santiago Ospina Idrobo
Universidad Eafit
Medellín, Colombia
sospinai@eafit.edu.co

Antoine Chavane de Dalmassy C.
Universidad Eafit
Medellín, Colombia
achavaned@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 $T(n) = 2 T(n-1) + C2$

$2*n \ (n>2) = 2*1 + 2*2 + 2*3 + 2*4 + 2*... + 2*100 + 2*... + 2*10000000000$

3.2

Este fue el código para probar el tiempo en el que se demora en calcular los diferentes tamaños de rectángulos:

```
public class PruebaRect {
    public static void main(String[] args) {
        PruebaPuertoAnt();
    }
    public static int formas(int n){
        if(n<=2){
            return n;
        }else{
            return formas(n-1)+(n-2);
        }
    }
    public static void PruebaPuertoAnt(){
        int[] tamagnos =
{2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,40,50};
        for(int i=0;i<tamagnos.length;i++){
            long prpio = System.currentTimeMillis();
            formas(tamagnos[i]);
            long end = System.currentTimeMillis();
            double rest = (double)(end - prpio);
            System.out.println("El tiempo para n = "+tamagnos[i]+" fue de
"+rest+" milisegundos.");
        }
    }
}
```

ESTRUCTURA DE DATOS 1 Código ST0245

```
    }
}
```

3.3 No, es demasiado lento.

3.4 El ejercicio groupSum5 nos dice que: dada una matriz de entrada, se puede elegir un grupo de entradas, de modo que el grupo sume el objetivo dado con las restricciones, todos los múltiplos de la matriz deben incluirse, pero si el valor que sigue luego de un múltiplo de 5 es 1, entonces este no se agrega.

Por ejemplo: si empezamos desde un entero: 0, nos dan una matriz de entrada: [2,5,10,4] y el numero al que debe dar la respuesta al ser sumados es 19; entonces el programa nos retornara verdadero, ya que 5 y 10 son múltiplos de 5, por lo cual se deben sumar y ahí obtenemos 15 y con el número de la tercera posición del arreglo que es 4, la suma nos daría 19 por lo que es verdadero.

El código para la solución de este problema es el siguiente:

```
public boolean groupSum5(int start, int[] nums, int target) {
    if (start >= nums.length)
        {
            if (target == 0)
                return
true;
            return
false;
        }
    if(nums[start] % 5 == 0)
    {
        if (start < nums.length - 1 && nums[start+1] == 1)
            return groupSum5(start + 2, nums, target - nums[start]);
        return groupSum5(start + 1, nums, target - nums[start]);
    }
    if (groupSum5(start + 1, nums, target - nums[start]))
        return true;
    return groupSum5(start + 1, nums, target);
}
```

Explicación:

En el primer condicional la condición es saber: si el numero de comienzo es mayor o igual a el tamaño del arreglo nums, si cumple con esa condición luego entra a otro condicional para saber si la suma es 0 si este cumple retorna verdadero, sino retorna falso.

Si no cumple con el primer condicional entra a otro condicional, donde la condición que si el modulo de cinco de el numero que se encuentre en la posición del número registrado como start en 0, entonces entra a otro

ESTRUCTURA DE DATOS 1 Código ST0245

condicional si el anterior se cumple, donde este nuevo condicional mira si el número start es menor que el tamaño de el arreglo menos uno y la posición de el número start + 1 es igual a 1, si este se cumple se hace un llamado recursivo donde el numero start se le va a sumar, el arreglo sigue igual y el target se le resta el numero que este en la posición start de el arreglo, si no se cumple la condición entonces hace un llamado recursivo igual anterior solo que esta vez solo restándole - 1 a el start.

Si ninguno de los dos primeros condicionales importantes cumple entonces se entra a el último que seria un llamado recursivo de start +1, el arreglo igual y target menos el numero de la posición start del arreglo nums, si este se cumple es verdadero; sino se retorna un llamado recursivo de start + 1 el arreglo igual al igual que target.

Y con este código en el cual se emplea la recursión, podremos solucionar el problema planteado a el principio del ejercicio.

3.5

3.6 El cálculo de complejidad para algoritmos recursivos:

n: Es el número de llamadas recursivas generadas por el método.

C: Es el coste iterativo des resto de operaciones que realiza el método excluyendo la invocación de las llamas recursivas. i: Es el número total de llamadas recursivas.

4) Simulacro de Parcial

[opc]4.1 SumaGrupo(start + 1, nums, target);

4.2 a) $T(n) = T(n/2) + C$

4.3

4.5

4.5.1 Línea 2: n

Línea 3: n-1

Línea 4: n-2

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

4.6

4.6.1 R/: Línea 10: sumaAux(n, i+2)

4.6.2 R/: Línea 12: sumaAux(n, i+1)

[opc]4.7

4.7.1 R/: Línea 9: comb(S, i + 1, t - S[i])

ESTRUCTURA DE DATOS 1 Código ST0245

4.7.2 R/: Línea 10: `comb(S, i + 1, t)`

4.8

4.8.1 `return 0`

4.8.2 `ni + nj`

[opc]4.9 c). 22

4.10 b). 6

4.11

4.11.1 Línea 4: `n-1` , `lucas(n-2)`

4.11.2 c). $0(2^n)$

4.12

4.12.1 Línea 13: `sat`

4.12.2 Línea 17: `Math.max(fi + fj)`

4.12.3 Línea 18: `sat`

[opc]6 informe -

Laboratorio N1

viernes 16 de agosto de 2019

Se repartieron los puntos; la repartición fue:

Santiago: 1.2, 2.2, 3.1--3.4--3.5--3.6, 4.2--4.5--4.8--4.11

Antoine: 1.1, 2.1, 3.2--3.3--3.5--3.6, 4.3--4.6--4.10--4.12

Sábado 17 de agosto de 2019

Antoine realizo los puntos: 1.1, 2.1, 3.2--3.3

Santiago realizo los puntos: 1.2, 2.2, 3.1--3.4

Domingo 18 de agosto de 2019

Antoine corrección del punto: 1.2(Santiago), se corrigió un error en el código.

Antoine realizo los puntos: 4.3--4.6--4.10--4.12 [opc]: 4.9--6

Santiago realizo los puntos: 4.2--4.5--4.8--4.11 [opc]: 4.1—4.7