ESTRUCTURA DE DATOS 1
Código ST0245

Laboratory practice 1:

Tomás Calle
Universidad Eafit
Medellín, Colombia
tcallee@eafit.edu.co

Juan Camilo Salazar
Universidad Eafit
Medellín, Colombia
jcsalazaru@eafit.edu.co

Member	Date	Done	Doing	Todo list
tomas calle	23/02/2021	codingBat 2	Make the document, and organize everything so we can start to make the final paper.	Volume algorithm or 1.2
Juan Camilo salazar	23/02/2021	3 exercices of codingBat 2	finish recursion 1 of codingBat	make changes to the Genome algo.
tomas calle	25/02/2021	Volume 2 algo Finished	helping to debug the Genome algo	Start and finish the test Practice.
Juan Camilo Salazar	25/02/2021	.5 Test Practice	.5 test practice	make changes to the Genome algo

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

1.1

```java
public static int genoma(String cadena, String subCadena){
        if (cadena.length() == 0 || subCadena.length() == 0) {
            return 0;
        } else if (cadena.charAt(0) == subCadena.charAt(0)) {
             return 1 + genoma(cadena.substring(1), subCadena.substring(1));
        }
        int llamado1 = genoma(cadena.substring(1), subCadena);
        int llamado2 = genoma(cadena, subCadena.substring(1));
        if (llamado1 > llamado2) {
            return llamado1;
        } else {
            return llamado2;
        }
    }
```

1.2

```java
public static int fit1x2(int nrec ,int tiles){
        if(nrec*2 == tiles){
            return 1;
        } else if(nrec*2 < tiles){
            return 0;
        }else{
            return fit1x2(nrec, tiles + 2) + fit1x2(nrec, tiles + 4);
        }
    }
```

2.1

```java
1) public boolean array220(int[] nums, int index) {

        if(index>=nums.length-1){
          return false;
        }else if(nums[index]*10==nums[index+1]){
          return true;
        }
        else return array220(nums,index+1); // F(n-1)
      }
       O(n-1) n being the length of the static array.
```

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

2)
```java
public boolean nestParen(String str) {
    if(str.length()==0){
      return true;
    }
    else if(str.length()==1) {
      return false;
    }
    else if((str.charAt(0)!='(' && str.charAt(str.length()-1)==')')
    ||(str.charAt(0)=='(' && str.charAt(str.length()-1)!=')')
    ||(str.charAt(0)!='(' && str.charAt(str.length()-1)!=')')){
      return false;
    }
    return nestParen(str.substring(1,str.length()-1));
  }
```
O(n-2) = O(n) n being the length of the String

3)
```java
public String parenBit(String str) {
    if(str.length()==0) return "";
    else if(str.charAt(0) =='('&&str.charAt(str.length()-1)==')'){
       return str;
    }else if(str.charAt(0) == '('&&str.charAt(str.length()-1)!=')'){
      return parenBit(str.substring(0,str.length()-1));
    }else if(str.charAt(0) != '('&&str.charAt(str.length()-1)==')'){
      return parenBit(str.substring(1));
    }
    else{
      return parenBit(str.substring(1,str.length()-1));
    }
  }
```
O(n-2) = O(n) n being the length of the String

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

4)
```java
public int countAbc(String str) {
    if(str.length()<3){
     return 0;
    }
    if(str.substring(0,3).equals("abc")||str.substring(0,3).equals("aba")){
      return 1 + countAbc(str.substring(1));
    }else return countAbc(str.substring(1));
  }
```
O(n-2) = O(n) n being the length of the String

5)
```java
public int strCount(String str, String sub) {
    if(str.length()==0) return 0;
    else if(str.length()<sub.length()) return 0;
    else if(str.substring(0,sub.length()).equals(sub)){
      return 1+strCount(str.substring(sub.length()),sub);
    }else return strCount(str.substring(1),sub);
  }
```
O(n-2) = O(n) n being the length of the String

2.2

```java
public boolean groupSum6(int start, int[] nums, int target) {
        if(start>=nums.length){ // 3C_1
            return target == 0; // 2C_2
        }else if(nums[start] == 6){ // 4C_3
            return groupSum6(start+1,nums,target-6); // 4C_4 + F(n-1)
        }
        return groupSum6(start+1,nums,target-nums[start])||groupSum6(start+1,nums,target);
           // 7C_5 + F(n-1) + F(n-1)
    }

    // F(n) = c_1 + 2^(n - 1) (where c_1 is an arbitrary parameter)
    // O(2^(n))
    n = the number of elements in the array nums.

public boolean groupSum(int index,int[] nums,int target){
    if(index == nums.length){ // 3C_1
         return target == 0; // // 2C_2
```

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

```
        }else{
            return groupSum(index + 1,nums,target-nums[index]) || groupSum(index + 1,nums,target);
               // 7C_3 + F(n-1) + F(n-1)

               // F(n) = c_1 2^(n - 1) (where c_1 is an arbitrary parameter)
               //O(2^(n)) donde n es la longitud del arreglo

               n = the number of elements in the array nums.

        }
    }

public static boolean groupNoAdj(int index, int[] nums, int target){
        if(index >= nums.length){ // 3C_1
            return target == 0; // 3C_2
        }
        return groupNoAdj(index + 2,nums,target-nums[index]) || groupNoAdj(index + 1,nums,target);
        // 7C_3 + F(n-1) + F(n-1)
        // F(n) = c_1 2^(n - 1) (where c_1 is an arbitrary parameter)
        O(2^(n))


        n = the number of elements in the array nums.
    }

public static boolean groupSum5(int start, int[] nums, int target){

        if(start >= nums.length){
           return target == 0;
        }

        if(nums[start]%5 == 0){
            if(start < nums.length-1){
                if(nums[start+1] == 1){
                    return groupSum5(start+2,nums,target-nums[start]);
                }
                return groupSum5(start+1,nums,target-nums[start]);
            }
        }
        return groupSum5(start+1,nums,target-nums[start]) || groupSum5(start+1,nums,target);

        O(2^(n))
```

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

```
      n = the number of elements in the array nums.


   }
public static  boolean groupSumClump(int start, int[] nums, int target) {
       if(start >= nums.length){ // 3C_1
           return target == 0; // 2C_1
        }
        int countadorValor = 0; // C_2
        int countadorIndex = 0;   // C_3
        boolean seguidos = true; // C_4

        for(int tt = start; tt < nums.length;tt++){ // 2C_5 * F(n-1)
           seguidos = (nums[tt] == nums[start]);

            if(seguidos){
                countadorValor = countadorValor + nums[tt];
                countadorIndex++;
            }
        }
        return (
        groupSumClump(start+countadorIndex,nums,target-countadorValor)F(n-1)
        ||
        groupSumClump(start+countadorIndex,nums,target));F(n-1

    } //F(n) = F(n-1)+F(n-1)+F(n-1) ⟷ O(3^n)

      n = the number of elements in the array nums.
```

Input:

$$f(n) = f(n-1) + f(n-1) + f(n-1)$$

Result:

$$f(n) = 3\, f(n-1)$$

Recurrence equation solution:

$$f(n) = c_1\, 3^{n-1}$$ (where c_1 is an arbitrary parameter)

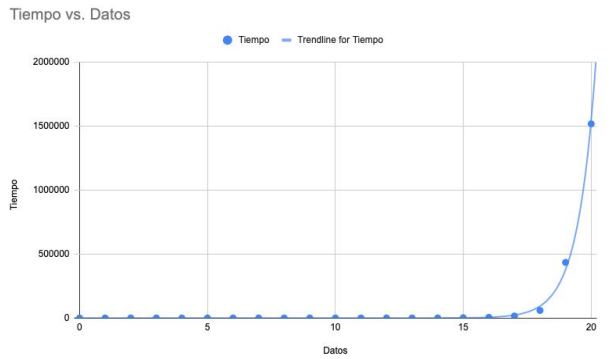
---

2 3.1. Calculen la complejidad, para el peor de los casos, del ejercicio 1.1

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

- O(2^n) al tener dos llamados recursivos

3.2. Analyzing the algorithm with 20 different cases.



Donde los datos es la cantidad de caracteres en cada string

3.3. ¿La complejidad del algoritmo del ejercicio 1.1 es apropiada para encontrar la subsecuencia común más larga entre ADNs mitocondriales como los de los datasets?

No es apropiada porque tiene complejidad de O(2^n) entonces se demorara varios minutos en analizar tanta información. Esto significa que el algoritmo es bastante ineficiente en términos de tiempo.

3.5. Calculate the complexity of the following exercises:2.1,2.2,3.6 and explain in your own words the meaning of n or m depending on the exercise.

The analysis of the complexities are in the photos above. (2.1,2.3,3.6)

4. Simulacro de Parcial

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

1. Completa la línea 6 .....
   - s.substring(0, i)
   - b. s.substring(0, n)
   - c. s.substring(i, n)
2. Completa la línea 9 .....
   - a. false
   - b. s.substring(0, i)
   - true
3. Completa la línea 11 .....
   - solve(t, s.substring(i), n - i)
   - b. solve(pfx, t), n - i)
   - c. solve(t, s.substring(n), I - n)

1. Completa la línea 9 .....
   Y Completa la línea 10 .....
2. Completa la línea 11 .....
   Y Completa la línea 12 .....
3. ¿Cuál es la ecuación de recurrencia que representa la complejidad, en el tiempo, para el peor de los casos, en términos de $p=n+m$?
   T(p) = ...

```
FloodFill Utill(screen, x+1, y+1,prevC,newC,N,M);
FloodFill Utill(screen, x-1, y-1,prevC,newC,N,M);
FloodFill Utill(screen, x+1, y-1,prevC,newC,N,M);
FloodFill Utill(screen, x-1, y+1,prevC,newC,N,M);
```

2-5 : C_1
9-12 : C_2 + f(m-1)^2*f(n-1)^2
12-16 : C_3 + f(m-1)2*f(n-1)^2
2-16 : C_1+C_2+C_3 + (2*F(m-1)+F(n-1))+(2F(m-1)+2*F(n-1))
= 4F(m-1)+4F(n-1)
F(m) = 4*F(m-1)
F(m) = 4 ^ m-1
F(n) = 4*F(n-1)
F(n) = 4 ^ n-1

O(4^m + 4^n)

La complejidad de la función *mistery* es

a. T(n,m) = C x n x m
- T(n,m) = C x n x m^2
c. T(n,m) = C x n x m.log m
d. T(n,m) = C x n x m^3

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

```
01 static boolean isPal(String s) {
02  if(s.length() == 0 || s.length() == 1)
03   return [ true ];
04  if( [ s.charAt(0) == s.charAt(s.length()-1) ])
05     return isPal(s.substring(1, s.length()-1));
06   //else
07  return false;
08 }
```

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473