

LABORATORIO 1

Santiago Albisser-Juan Pablo Leal

SIMULACRO DE PREGUNTAS SUSTENTACIÓN DE PROYECTOS

3.2-

```
public int strCount(String str, String sub) {
    if (str.length() < sub.length()){ //constante
        return 0; //constante
    }
    if(str.substring(0,sub.length()).equals(sub)){ //constante
        return 1+strCount(str.substring(sub.length()),sub); //constante+T(n-sub.length)
    }else{
        return strCount(str.substring(1),sub); //T(n-1)
    }
}

public int countAbc(String str) {
    if(str.length()<3){ //constante
        return 0; //constante
    }
    if(str.substring(0,3).equals("abc") || str.substring(0,3).equals("aba")){//constante
        return 1+countAbc(str.substring(1)); //constante+T(n-1)
    }else{
        return countAbc(str.substring(1)); //T(n-1)
    }
}

public String pairStar(String str) {
    if(str.length()<2){ //constante
        return str;
    }
}
```

```

    if(str.charAt(0)==str.charAt(1)){//constante
        return str.charAt(0)+"*"+pairStar(str.substring(1));//constante+T(n-1)
    }else{
        return str.charAt(0)+pairStar(str.substring(1));//constante+T(n-1)
    }
}

public int countX(String str) {
    if (str.length()==0){//constante
        return 0;//constante
    }
    if (str.charAt(0)=='x'){//constante
        return 1+countX(str.substring(1));//constante+T(n-1)
    } else {
        return countX(str.substring(1));T(n-1)
    }
}

public int count8(int n) {
    if (n==0){//constante
        return 0;//constante
    } else if (n>=88&& n%100==88){//constante
        return 2+count8(n/10);//constante+T(n/10)
    } else if (n%10==8){//constante
        return 1+count8(n/10);//constante+T(n/10)
    } else {
        return count8(n/10);//T(n/10)
    }
}
}

```

3.3-

N representa el numero de valores a los que se le esta haciendo la recursión. Dependiendo de cada ejercicio, puede ser un arreglo, el tamaño de un string, etc.

3.4-

Stack overflow sucede cuando un programa en el computador intenta usar mas memoria de la que tiene programada. Puede suceder cuando se pretenden hacer operaciones con números muy grandes.

3.5-

El valor más grande que se pudo sacar de la serie de fibonnacci fue de 50 y además se logro debido al procesador del computador en el que lo realizamos. Para ejecutarlo con un millón se necesitaría un procesador mas potente para almacenar mas operaciones y memoria.

SOLUCIÓN SIMULACRO PARCIAL.

1-Start+1,nums,target.

2-A

3-

3.1) (n-a, b, c)

3.2) res, solucionar (n-b, a, b, c) +1

3.3) res, solucionar (n-c, a, b, c) +1

5-

5.1)

Línea 2= return n;

Línea 3= n-1

Línea 4= n-2.

5.2) B

6-

6.1) Linea 10= return (sumaAux(n,i+2));

6.2) Linea 12= return (n.charAt(i) - '0') + sumaAux(n,i+1);

8-

8.1) Linea 9=retun 0;

8.2) Linea 13=ni+nj;