PARCIAL TEORICO

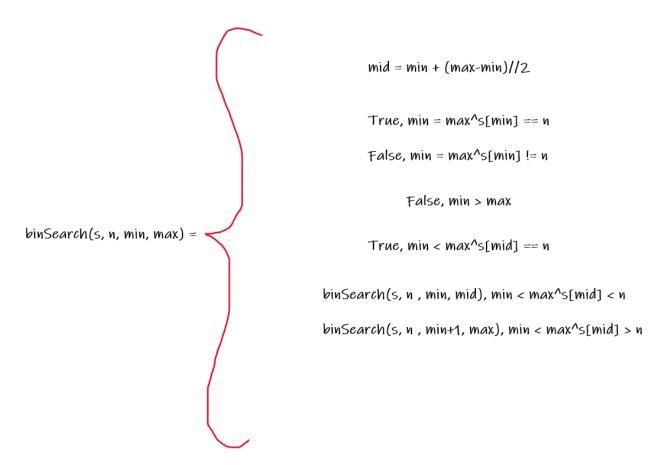
PRESENTADO POR: JUAN CAMILO BAZURTO ARIAS

PRESENTADO A: SEBASTIAN CAMILO MARTINEZ REYES

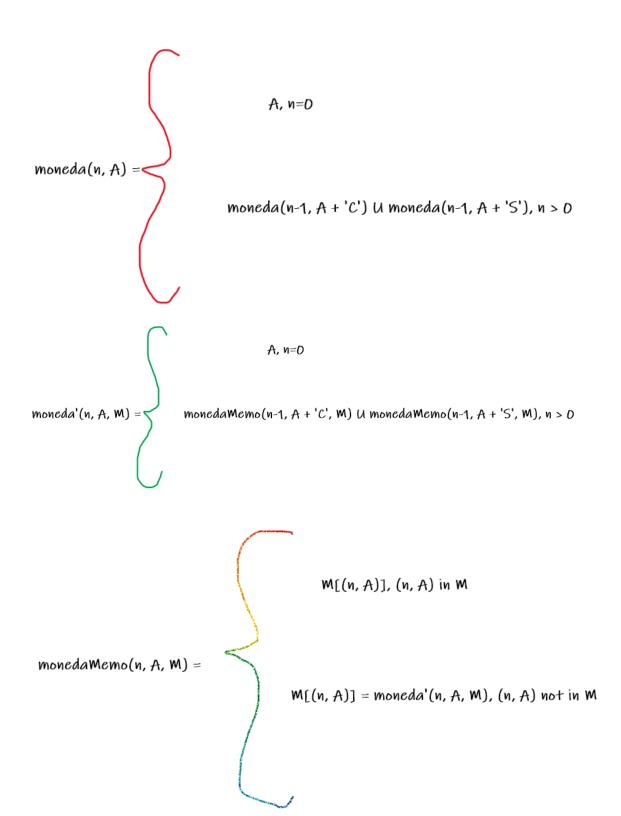
ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO ALGORITMOS Y ESTRUCTURAS DE DATOS PROGRAMA DE INGENIERÍA DE SISTEMAS BOGOTÁ D.C.

2021 - 1

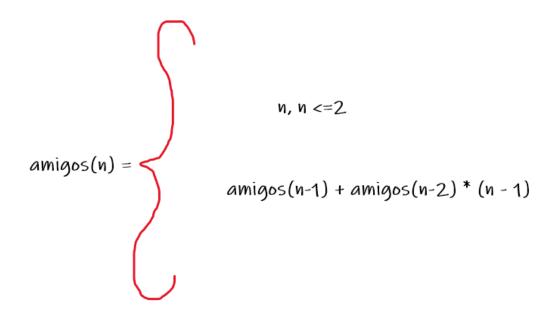
1. Dada una lista ordenada de elementos $A = x0, x1, x2 \dots xi \dots xn$. Escriba una función recurrente utilizando la estrategia de dividir y conquistar, donde dado un elemento q. La función responda si el elemento q se encuentra en la lista A.



2. Escriba una función f(n . . .) recurrente que satisfaga lo siguiente: Suponga que Juan lanza una moneda al aire N veces, cuyos valores posibles son 'C'(Cara)y 'S'(S el 1 o) para cada evento independiente de lanzamiento. Pedro dese a llevar registro de todos los eventos de lanzamiento que produce Juan. Por lo que para el caso de N= 3, El registro de pedro es el conjunto A={CCC, CCS, CSC, CSS, SCC, SCS,SSC, SSS}, escriba una función que genere el conjunto A para cualquier N. Aplique el método de memorización por Top-Down^



3. Suponga que hay n amigos, quienes pueden permanecer solos o ser emparejados con otro amigo. Cada amigo puede ser emparejado una sola vez. Encuentre el número de formas en que los amigos pueden quedar solos o emparejados.



4. Ordene de 1 a N las siguientes funciones características de crecimiento asintótico.

f(x)	Orden
n^k	5
nlg(n)	3
n^2	4
ln(n)	1
n	2

- 5. Marque la casilla, solo si la afirmación que la acompaña es verdadera.
 - $\Omega(n) = \{f(n) : \text{existen constantes positivas c y n0tales que } 0 \le f(n) \le cg(n) \text{ para todo } n >= n0$

Falso, porque la función omega representa el mejor de los casos, por lo tanto, la función que tiene la constante debería estar $f(n) \le 0$

• El teorema maestro aplica para la solución de recurrencias de la forma T(n) = aT(n/b) + f(n)

Verdadero, por definición de teorema maestro.

• Un algoritmo se considera correcto si para un subconjunto de la entrada. Entrega una salida correcta.

Falso, porque el algoritmo debe retornar todo el problema y no solo una solución parcial.

6. Dado el algoritmo de insertion-sort realice una estimación del tiempo de ejecución en donde demuestre que T(n) es de la forma T(n) = an2 + bn + c

```
Algorithm 1 Insertion-Sort
 1: procedure INSERTION-SORT(A, n)
 2:
       inversions = 0
                                                   C1
       L[0..n-1] be a new array.
                                                   C2
                                                              1
 3:
                                                   C3
                                                              1
       for q=0 do n-1
 4:
                                                   C4
                                                              n-1
          L[q] = A[q]
 5:
 6:
       end for
                                                   C5
                                                               0
       for j=1 do n-1
                                                   C6
                                                               n-1
 7:
                                                   C7
                                                               n-2
          key = L[j]
 8:
                                                   C8
          i = j - 1
                                                               n-2
 9:
           while i \geq 0 and L[i] > key do
                                                   C9
                                                               (n-2)*(n-1)
10:
                                                  C10
                                                            (n-2)*(n-2)
              L[i+1] = L[i]
11:
                                                  C11
                                                             (n-2)*(n-2)
              j = i - 1
12:
              inversions = inversions + 1
                                                             (n-2)*(n-2)
                                                  C12
13:
           end while L[i+1] = key
14:
       end for
15:
16:
                                                  C13
                                                           1
       return inversions
17:
18: end procedure
```

$$T(n) = (a)n^2 + bn + c$$