William Mendez – 202012662

Juliana Galeano – 202012128

Daniel Aguilera – 202010592

Boris Reyes – 202014743

TAREA 4

Tarea IV: parte I Para los siguientes problemas diseñe el algoritmo (GCL con pre y postcondición), seleccione estructuras de datos a usar y especifique su complejidad en tiempo (exacta y asintótica).

1. Dado un conjunto de números entregados en forma de arreglo de tamaño n, determine el elemento que es la moda (elemento que ocurre el mayor número de veces).

```
fun modaHallar ( A: array of int(0,N) ) : int
    var  M : HashMap of int, int
    var S : Array of int
    var i, j, k, mayor: int
    {Q:true}
    i,j,k,moda;=0,0,0;  → c1
    do i< N →            → c2(n+1)
        if M.containsKey(A[i]) = false → M.put(A[i],1);  → n(c4 + c5 + c2 + c6 + c4)
        [] M.containsKey(A[i]) = true  → j := M.get(i) + 1;→ n(c5 +c2 + c1 + c2 + c7)
                                        M.put(A[i], j); n(c6 + c4)
        fi
        i := i+1;  →n(c1+c3)
    od

    S:=M.keySet() → c8
    do k< N → → c2(n+1)
        if M.get(S(k)) > M.get(moda) → moda := S(k); n(c7+c4+c7+c1+c4)
        fi
        k := k+1; n(c1+c3)
    od
{R:""}
```

ret moda

- Complejidad exacta
  c4 = acceder a un elemento de un array
  c5 = containsKey hashmap
  c6 = put hash map
  c7 = get hash map
  c8 = keyset hashmap

c1 + c2 + c2 + **n(c4 + c5 + c2 + c2 + c6 + c4) + n(c5 +c2 + c1 + c2 + c7) + n(c6 + c4) + n(c1+c3) + c8 + c2(n+1) + n(c7+c4+c7+c1+c4) + n(c1+c3)**

= c1 + 2c2 + c8 + n(5c2+4c4+2c5+2c6+2c3+3c1)

=k1+nk2

=O(n)

- Complejidad Asintótica

Es posible ver que el algoritmo puede ser $\theta(n)$, solo basta con encontrar un c>k2 y un c que podría ser c = 1/k2 para que pueda ser ambas.

Con este tenemos que es $\theta(n)$.

2. Dado un conjunto de números entregados en forma de arreglo de tamaño n, determine el elemento que es la mediana (si hay un número par de datos, la mediana es el promedio de los dos números medios).

fun medianaHallar (A: array of int(0,N)) : float

var M : array of int

var length, indexMediana: int

var mediana: float

{Q:true}

length,indexMediana;=0,0; → **c1**

M := Mergesort (A);   → **n(log(n))**

length := A.length → **c1 + c3**

indexMediana := (length - 1) intdiv 2   → **c1+2c3**

if length mod 2 = 0 → mediana:=M[indexMediana] → **c2+ C1+c3**

[] length mod 2 ≠ 0 → mediana:=(M[indexMediana] + M[indexMediana+1]) div 2 → **c2+ C1+ 3c3**

fi

{R:""}

ret mediana

$c1+ n(\log(n))+c1+c3+c1+2c3+c2+c1+c3+c2+c1+3c3$

$= (5c1+2c2+7c3)+ n(\log(n))$

$=k+n(\log(n))$

$=O(n(\log(n))$

3. Diseñe (3) algoritmos para determinar si dos conjuntos (de tamaño m y n) son disjuntos. Analizando la complejidad de los algoritmos en términos de m y n, deberían tener las siguientes complejidades:

- $O\ n \log(n) + mlog(n)$

```
fun conjuntosDisjuntosv1(c1: set of int(0,n), c2 :set of int(0,m)): boolean
        var t: AVL tree of int
        var a1, a2: Array of int
        var i: int
        var resul: boolean
        a1, a2 := c1.toArray(), c2.toArray()
        resul := false
        i := 0
        do i<n →
                t.insert(a1[i])
                i := i+1
        i := 0
        do i<m ∧ !encuentra →
                resul = t.contains(a2[i])
                i := i+1
        return resul
```

- $O\ n \log(m) + mlog(m)$

```
fun conjuntosDisjuntosv2(c1: set of int(0,n), c2 :set of int(0,m)): boolean
        var t1, t2: AVL tree of int
        var a1, a2: Array of int
        var i: int
        var resul: boolean
        a1, a2 := c1.toArray(), c2.toArray()
```

```
resul := false
i := 0
do i<m →
    t.insert(a2[i])
    i := i+1
i := 0
do i<n ∧ !encuentra →
    resul = t.contains(a1[i])
    i := i+1
return resul
```

- $O(n)$ en el caso esperado.

Pista: Use tabla hash y analice una posible relación de orden entre n y m.

Tarea IV: parte II ▪ Resuelva las siguientes ecuaciones de recurrencia usando el método maestro:

- $Tn = 8T(n/2) + \Theta(n^2)$

  $\mathrm{f}(n) = \mathrm{O}\left(n^{(log_2 8)}\right)$

  $\mathrm{f}(n) = \mathrm{O}(n^3)$

  Caso 1, $\in = 1$

  $T(n) = \Theta(n^3)$

- $T(n) = 3T(n/2) + nlog(n)$

  $\mathrm{f}(n) = \mathrm{O}\left(n^{(log_2 3)}\right)$

  $\mathrm{f}(n) = \mathrm{O}(n^{1.58})$

- $\mathrm{T}(n) = 4\mathrm{T}(n/2) + n^2\sqrt{n}$

  $\mathrm{T(n) = 4T(n/2) + n^{2+1/2}}$

  $\mathrm{f}(n) = \mathrm{O}\left(n^{(log_2 4)}\right)$

  $\mathrm{f}(n) = \mathrm{O}(n^2)$

  Caso 3, $\in = ½$

  $4\left(\frac{n}{2}\right)^{(2+1/2)} \leq c \cdot n^{(2+1/2)}$

  $c \geq \frac{\sqrt{2}}{2} \geq 1$

  $T(n) = \theta\left(n^{2+1/2}\right)$