

Primer Examen Parcial Técnicas y Prácticas de Programación

Prof. Luis Garreta
Ingeniería de Sistemas y Computación
Pontificia Universidad Javeriana – Cali

4 de septiembre de 2017

1. Una arreglo es un **isomorfo** de otro arreglo si el segundo arreglo contiene todos los elementos del primer arreglo. Por ejemplo, A: [1,7,9,0] es un isomorfo de [7,1,0,9]. El siguiente algoritmo resuelve el problema

- Cual es la complejidad en el peor caso? Argumente porqué
- Cual es la complejidad en el mejor caso? Argumente porqué
- Se puede resolver este problema por divide y vencerás? Argumente porque si o porque no.

```
int Isomorfos(int *arr1, int *arr2,
             int n1, int n2) {
    if(n1 == n2) {
        for(int i = 0; i < n1; i++) {
            bool found = 0;
            int j = 0;
            while(!found && j < n1) {
                if(arr1[i] == arr2[j])
                {
                    found = 1;
                    arr2[j] = '#';
                }
                j++;
            }
            if(!found)
                return 0;
        }
    }
    else {
        return 0;
    }
    return 1;
}
```

2. Desarrollar una función en C++ que retorne el número de números negativos (menores a cero) dentro de un arreglo **A** de números enteros de tamaño n . Por ejemplo, el arreglo $A = [2, 1, -3, 4, -2, 1, 9, -2]$, de tamaño 8 retornará 3, ya que existen 3 números negativos. Asuma que el tamaño n es potencia de 2 y que existe la función *split*:

int *split (int arreglo [], int inicio, int final);

que dado como entrada un arreglo A y dos enteros *inicio* y *final*, retorna un nuevo arreglo con los elementos del arreglo desde el inicio hasta el *final*, sin incluirlo.

- Implemente la función en C++.
 - Calcule la complejidad y muestre el proceso como la obtiene.
3. Muestre cual es la complejidad para las siguientes ecuaciones de recurrencia:
- $T(n) = 8T(\frac{n}{2}) + n^2$
 - $T(n) = 8T(\frac{n}{2}) + n^3$
 - $T(n) = 8T(\frac{n}{2}) + n$
4. Construya el árbol completo que realizaría mergesort al ordenar el siguiente arreglo e indique en el árbol que etapa se está realizando en cada paso.

A:

10	25	77	12	9	13	4	2
----	----	----	----	---	----	---	---