

Universidad Mariano Gálvez de Guatemala  
Sede Boca del Monte  
Facultad Ingeniería en Sistemas de Información  
Ingeniero Melvin Cali  
Programación III



Métodos de Ordenamiento  
Y  
Funcionamiento del Programa

Integrantes

Nombres	Carnet
Ángel Enrique Ibañez Linares	7690-22-19119
Edras Fernando Tatuaca Alvarado	7690-22-11542
Jose Daniel Tobar Reyes	7690-21-13125
María José Macz López	7690-22-13438

## Métodos de Ordenamiento

### Buble Sort (Burbuja)

Es un método caracterizado por la comparación e intercambio de pares de elementos hasta que todos los elementos estén ordenados. En cada iteración se coloca el elemento más pequeño (orden ascendente) en su lugar correcto, cambiándose además la posición de los demás elementos del array. La complejidad del algoritmo es  $O(n^2)$ .

### Insertion Sort (Inserción)

Método usado para ordenar una mano de naipes. Los elementos están divididos conceptualmente en una secuencia destino y una secuencia fuente. En cada paso, comenzando con  $i=2$  e incrementando  $i$  en uno, el elemento  $i$ -ésimo de la secuencia fuente se toma y se transfiere a la secuencia destino insertándolo en el lugar adecuado.

### Merge Sort (Combinación)

Es el más común para ordenar datos utilizando la técnica de " *divide y vencerás* ". En este algoritmo, el problema se divide en subproblemas y luego, después de ordenarlos, se fusionan. Digamos que tenemos una lista de números sin ordenar, el resultado de una clase para un tema determinado. Para ordenarlos en orden ascendente necesitaremos colocarlos en una lista comenzando de menor a mayor.

### Quick Sort (Rápida)

Se basa en el hecho que los intercambios deben ser realizados preferentemente sobre distancias grandes. El algoritmo (técnica de dividir y vencer) simplificado es: Seleccionar un elemento del array (elemento pivote, p.e. el que se encuentra en la mitad). Todos los elementos menores al pivote se colocan en un array y los mayores en otro. Se aplica el mismo procedimiento de forma recursiva, sobre los subarrays hasta que solo exista un elemento.

### Heap Sort (Montón)

Heap es una estructura de datos de árbol especializada. El montón comprende el nodo superior llamado raíz (padre). Su segundo hijo es el hijo izquierdo de la raíz, mientras que el tercer nodo es el hijo derecho de la raíz. Los sucesivos nodos se rellenan de izquierda a derecha. La clave del nodo padre compárese al de su descendencia, y se produce un arreglo adecuado. El árbol es fácil de visualizar donde cada entidad se denomina nodo. El nodo tiene claves únicas para su identificación.

### Counting Sort (Conteo)

El algoritmo de ordenación por conteo funciona creando primero una lista de los conteos u ocurrencias de cada valor único en la lista. Luego crea una lista ordenada final basada en la lista de conteos. Una cosa importante que debes recordar es que la ordenación por conteo solo se puede usar cuando conoces de antemano el rango de valores posibles en la entrada.

## **Radix Sort (Raíz)**

Consiste en ordenar los números tomando en cuenta el valor relativo que tienen las cifras o dígitos de un número en un determinado sistema de numeración. La característica de este algoritmo está en que no hace comparaciones para ordenar las listas, simplemente se encarga de ir contando o agrupando los números que tengan el mismo valor relativo en determinada cifra.

## **Bucket Sort (Cubo)**

Es un algoritmo de ordenación que funciona dividiendo un vector en un número finito de recipientes. Cada recipiente es entonces ordenado individualmente.

El ordenamiento por casilleros (bucket sort en inglés) es un algoritmo de ordenamiento que distribuye todos los elementos  $n$  a ordenar entre un número  $n$  de casilleros. Cada casillero sólo puede contener los elementos que cumplan unas determinadas condiciones. Las condiciones deben ser excluyentes entre sí, para evitar que un elemento pueda ser clasificado en dos casilleros distintos. Después cada uno de esos casilleros se ordena uno por uno con otro algoritmo de ordenación (para que distinto según el casillero), o se aplica recursividad en este algoritmo para obtener casilleros con menos elementos.



## **Funcionamiento de Programa**

Ejecución del ingreso del programa, se realizó un menú en el cual se podrá escoger el método de ordenamiento a utilizar:

Selecion el metodo que desea utilizar

1. Selecion Sort
2. Bubble Sort
3. Insercion Sort
4. Merge Sort
5. Quick Sort
6. Heap Sort
7. Counting Sort
8. Radix Sort
9. Bucket sort
10. Salir

Ingresar los valores separados por coma , y seleccionamos el método de ordenamiento número 1 (Selecion sort)

Ingrese los valores separados por coma  
8,6,3,4,5

5  
6  
3  
4  
8

Y luego seleccionamos la opción 10. Salir

Y así sucesivamente ingresamos los demás métodos

5. Salir  
2

Ingrese los valores separados por coma  
8,6,3,4,5

Valores ordenados

3  
4  
5  
6  
8

Bubble Sort (Burbuja)

3

Ingrese los valores separados por coma

8,6,3,4,5

Valores ordenados

3

4

5

6

8

Insertion Sort (Inserción)

4

Ingrese los valores separados por coma

8,6,3,4,5

Valores ordenados

3

4

5

6

8

Merge Sort (Combinación)

5

Ingrese los valores separados por coma

8,6,3,4,5

Valores ordenados

3

4

6

5

8

BUILD SUCCESSFUL (total time: 8 seconds)|

Quick Sort (Rápida)



```
6
Ingrese la cantidad de datos a ordenar:
5
Ingrese el numero 1:
8
Ingrese el numero 2:
6
Ingrese el numero 3:
3
Ingrese el numero 4:
4
Ingrese el numero 5:
5
Arreglo ordenado:
3 4 5 6 8 BUILD SUCCESSFUL (total time: 33 seconds)
|
```

Heap Sort (Montón)

```
7
Ingrese los valores separados por coma
8,6,3,4,5
```

Valores ordenados

```
3
4
5
6
8
```

Counting Sort (Conteo)

```
8
Ingrese los valores separados por coma
8,6,3,4,5
```

Valores ordenados

```
3
4
5
6
8
```

BUILD SUCCESSFUL (total time: 6 seconds)

Radix Sort (Raíz)

9

Ingrese la cantidad de datos a ordenar:

5

Ingrese el dato 1:

0.8

Ingrese el dato 2:

0.6

Ingrese el dato 3:

0.3

Ingrese el dato 4:

0.4

Ingrese el dato 5:

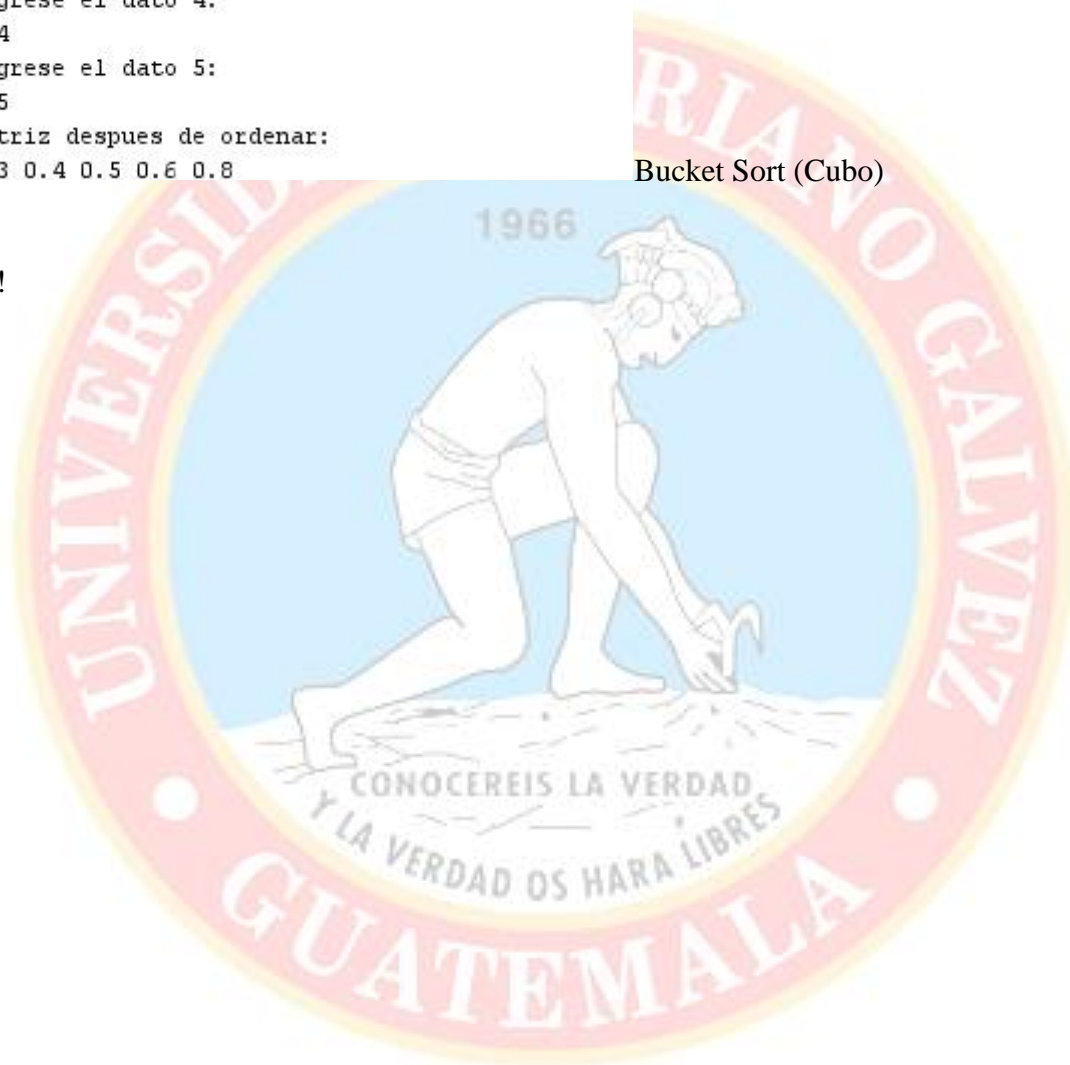
0.5

Matriz despues de ordenar:

0.3 0.4 0.5 0.6 0.8

Bucket Sort (Cubo)

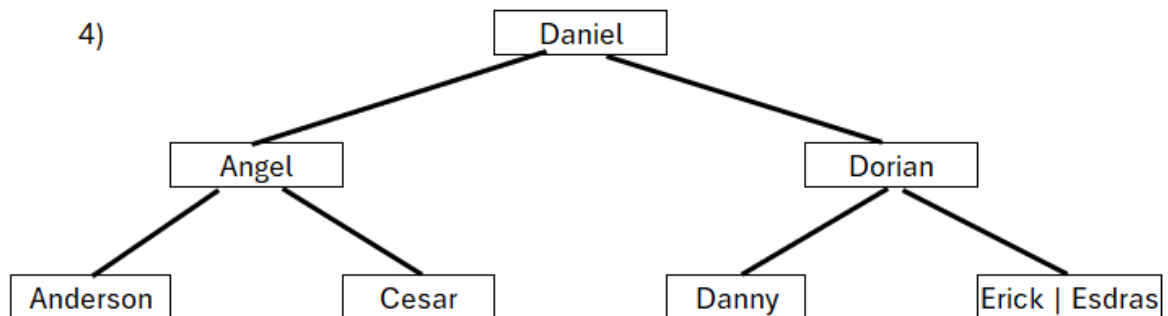
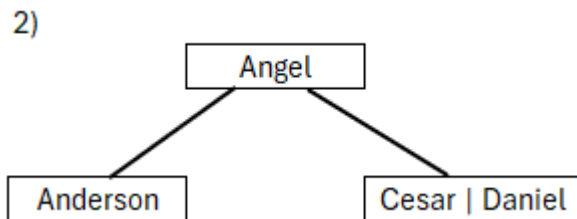
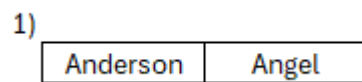
Fin!



## TAREA II PROGRAMACION III

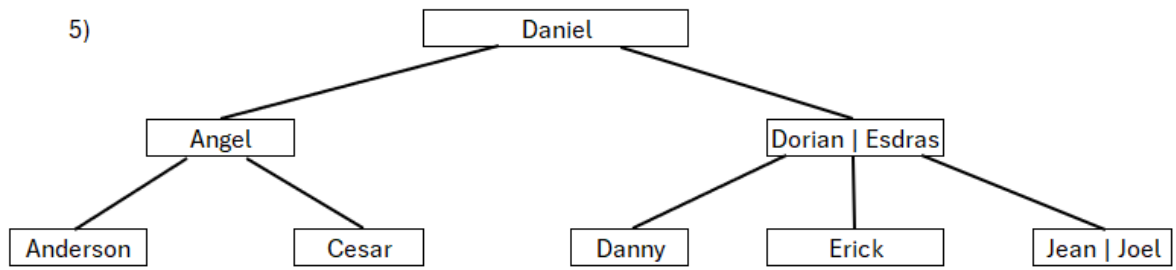
Realizar el árbol B de los siguientes nombres

Anderson	Danny	Jean	Maria	Susana
Angel	Dorian	Joel	Pablo	Victor
Cesar	Erick	Jonathan	Raul	Wilmer
Daniel	Esdras	Jose	Rocio	

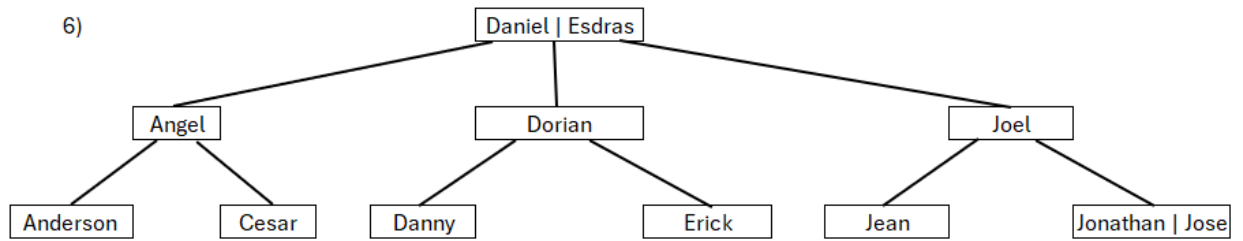




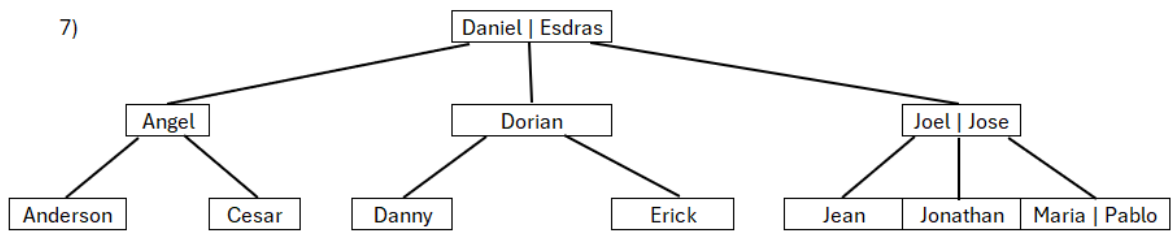
5)



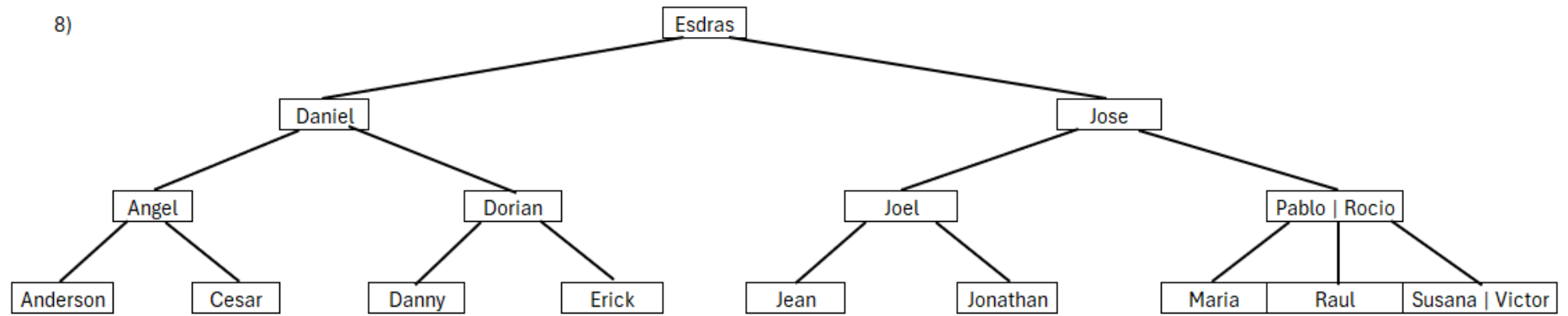
6)



7)



8)



9) Árbol Final

