```
#include <iostream>
using namespace std;
//Función para Generar numeros random en un rango Spesifico.
int numeroAleatorio(int min, int max){
    return min + rand()%(max - min + 1);
    }
Void imprimirArreglo(int arr[], int n){
        Const int rango=200;
            int bucket[rango]={0};
            for(int i=0;i<n; i++){
                bucket[arr[i]++;
            }
          int index=0;
        for(int i=0; i<rango; i++){
            while(bucket[i]>0){
                arr[index++]=i;
                bucket[i]--;
            }
        }
    }
int main(){
Const int Cantidad=10;
int arr[Cantidad];
Srand(time(0));
for(int i=0; i<Cantidad; i++){
            arr[i]=numerosAleatorios(0), 200);
            }
        Cout<<"Arreglo original:"<<endl;
```

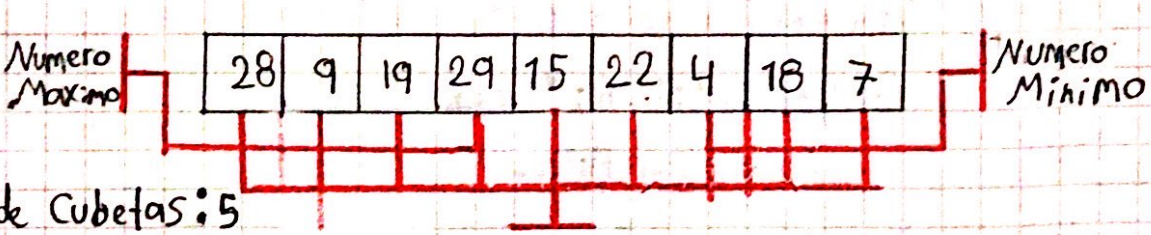
Scribe

Powered by CamScanner

```
    imprimerArreglo(arr, Cantidad);
    bucketSort(arr, Cantidad);
    Cout<<"Arreglo Ordenado:"<<endl;
    imprimirArreglo(arr, Cantidad);
    return 0;
}
```

Ordenamiento por Cubetas (bucket Sort)

- Consiste en tener un arreglo de n elementos:

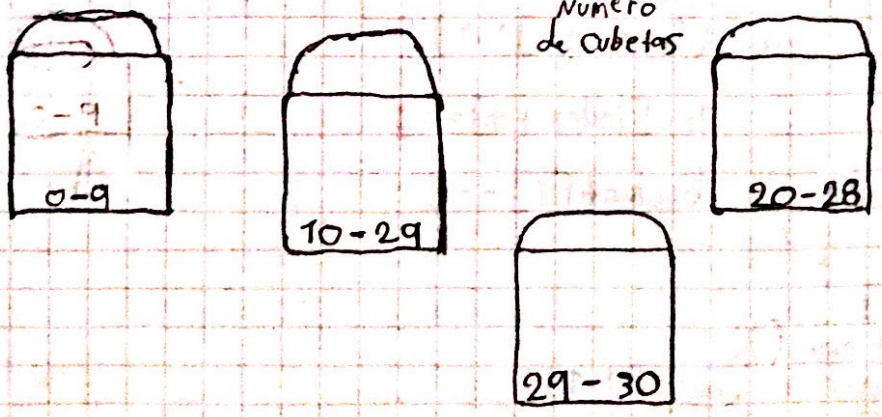


Numero de Cubetas: 5

Rango = Numero Maximo − Numero Minimo

- Ahora Tenemos que hayar El rango de cubetas para hacer ordenarlo por casillas o por cubetas y ordenarlos de acuerdo al rango:

$$\text{rango de cubetas} = \frac{\text{Rango}}{\text{Numero de cubetas}} = 4$$



Luego de ordenar los por rango las ordenamos otravez pero esta vez de menor a mayor de acuerdo al ranso de cubetas y haciendo el ordenamiento por cubetas.

Scribe
Powered by CamScanner