

Time Analysis - Análisis de Tiempos

Ejercicios



Suma y Promedio de números

```
int s = 0;
double prom = 0;
for (int i=0; i<n; i++)
{
    s = s + A[i];
    prom = s / (double)n;
}
cout<<Suma<<Prom<<" Promedio"<<prom;
```

Suma y Promedio de números

```
int s = 0;
double prom = 0;
for (int i=0; i<n; i++)
{
    s = s + A[i];
}
prom = s / (double)n;
cout<<Suma<<Prom<<" Promedio"<<prom;
```

Ordenamiento N números del 1 al 100

```
for (int i=0; i<n-1; i++)  
{  
    for (int k=i+1; k<n; k++)  
    {  
        if (vec[i] > vec[k])  
        {  
            int aux = vec[i];  
            vec[i] = vec[k];  
            vec[k] = aux;  
        }  
    }  
}
```

Ordenamiento N números del 1 al 100

```
int frec[101]={0};
for (int i=0; i<n; i++)
{
    frec[vec[i]]++;
}
int pos = 0;
for (int i=0; i<101; i++)
{
    for (int k=0; k<frec[i]; k++)
    {
        vec[pos] = i;
        pos++;
    }
}
```

Algoritmo raro - Infinito

```
int i=0;
int sum=0;
while (i < 100)
{
    if (i % 2 == 0)
    {
        for (int k=0; k<n; k++)
        {
            sum += vec[i];
        }
    }
    else
    {
        for (int k=0; k<i; k++)
        {
            sum += vec[i];
        }
    }
}
```

Factorial

```
int fact = 1;  
for (int i=2; i<n; i++)  
{  
    fact *= i;  
}  
printf("Factorial: %d", fact);
```

Buscar cadena de máximo 50 caracteres

```
int pos = -1;
for (int i=0; i<n; i++)
{
    if (strcmp(vec[i], cadBuscar) == 0)
    {
        pos = i;
        break;
    }
}
```


Buscar el mayor

```
int pos = 0;
for (int i=1; i<n; i++)
{
    if (vec[i] > vec[pos])
        pos = i;
}
printf("El mayor es: %d", vec[pos]);
```

Otro algoritmo raro – El while es válido?

```
int max = 0;
for (int m=0; m<n; m++)
{
    int cont = 0;
    int k = m + 1;
    while (vec[m] <= vec[k])
    {
        k = k + 1;
        cont++;
    }
    if (cont > max)
        max = cont;
}
printf("Maximo %d", max);
```

Dependiendo de los valores del vector este algoritmo puede ser que termine o no.

Logarítmica

```
int i=1;  
while (i < n)  
{  
    if (vec[i] % 2 == 0)  
        i *= 3;  
    else  
        i *= 2;  
}
```

Logarítmica 2

```
int i=1;
while (i < n)
{
    if (vec[i] % 2 == 0)
        n = n / 3;
    else
        n = n / 2;
}
```

Busqueda binaria – Arreglo ordenado

```
int inf = 0;
int sup = n - 1;
int pos = -1;

while ( (pos == -1) && (sup >= inf) )
{
    int medio = (inf + sup) / 2;
    if (arreglo[medio] == 80)
        posicion = medio;
    else if (arreglo[medio] < 80)
        limiteInferior = medio + 1;
    else
        limiteSuperior = medio - 1;
}
```

Implemente y analice las siguientes funciones

- Encontrar el numero mayor en un arreglo de enteros
- Ordenar un arreglo de numeros enteros
- Eliminar el elemento en una posicion de un arreglo
- Buscar un numero en un arreglo
- Calcular el factorial de N
- Determinar si un numero existe en un arreglo de enteros.
- Calcular cuantas veces se repite un numero X en un arreglo de enteros.
- Sumar los digitos de un numero entero positivo.
- Determinar si un numero es primo o no.
- Determinar la cantidad de primos que existen en un arreglo de enteros.