

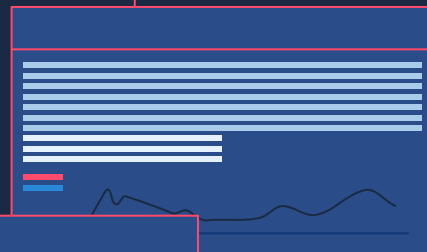


Tecnicatura en Programación Universitaria

Comisión

B

Clase 07

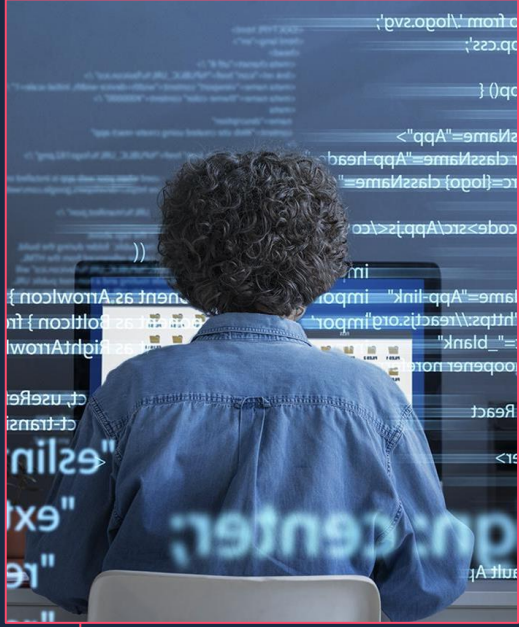
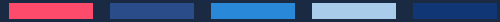


Clase Practica

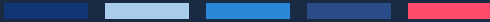
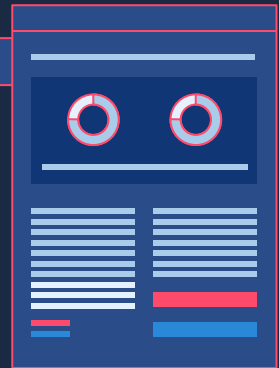
Martes 21 de Mayo 2024

Ing. Oviedo Codigoni Carlos Nicolas





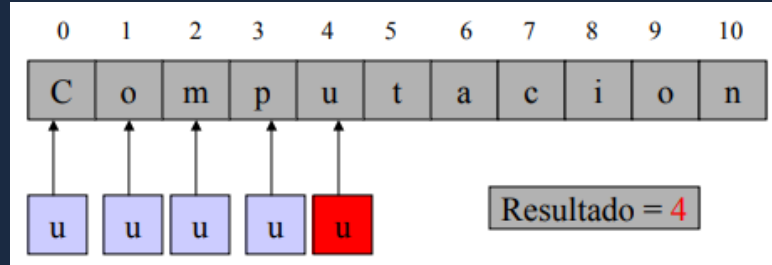
... Vectors & Matrices ...



Busqueda Secuencial

Consiste en ir comparando el elemento que se busca con cada elemento del arreglo hasta cuándo se encuentra.

Busquemos el Elemento 'u'



Búsqueda del Menor

```
menor = a[0];  
for (i=1;i<n;i++)  
    if ( a[i]<menor )  
        menor=a[i];
```

Búsqueda del Mayor

```
mayor= a[n-1];  
for (i=0;i<n-1;i++)  
    if ( a[i]>mayor )  
        mayor=a[i];
```

Búsqueda de Elemento

```
encontrado=-1;  
for (i=0;i<n;i++)  
    if ( a[i]==elemento_buscado )  
        encontrado=i;
```

Busqueda Binaria

La Búsqueda Binaria, compara si el valor buscado está en la mitad superior o inferior. En la que esté, subdivido nuevamente, y así sucesivamente hasta encontrar el valor. (El vector debe estar ordenado)

$(0+11)/2=5$		$(6+11)/2=8$		$(6+7)/2=6$		$(7+7)/2=7$	
0	2	0	2	0	2	0	2
1	3	1	3	1	3	1	3
2	4	2	4	2	4	2	4
3	5	3	5	3	5	3	5
4	6	4	6	4	6	4	6
5	8	5	8	5	8	5	8
6	13	6	13	6	13	6	13
7	17	7	17	7	17	7	17
8	19	8	19	8	19	8	19
9	23	9	23	9	23	9	23
10	25	10	25	10	25	10	25
11	26	11	26	11	26	11	26

Diagram illustrating the binary search process on a sorted array [2, 3, 4, 5, 6, 8, 13, 17, 19, 23, 25, 26] for the target value 17. The array is divided into four segments, each showing the current search range (L to U) and the midpoint calculation. The search proceeds by comparing the target value with the midpoint element and adjusting the search range accordingly.

Segment 1: $(0+11)/2=5$. Midpoint is 5 (value 8). $17 \leq 8$ is false, so the search range is updated to $L=6$. $17 > 8$ is true, so the search range is updated to $L=6$.

Segment 2: $(6+11)/2=8$. Midpoint is 8 (value 19). $17 \leq 19$ is true, so the search range is updated to $L=6$. $17 > 19$ is false, so the search range is updated to $L=8$.

Segment 3: $(6+7)/2=6$. Midpoint is 6 (value 13). $17 \leq 13$ is false, so the search range is updated to $L=7$. $17 > 13$ is true, so the search range is updated to $L=7$.

Segment 4: $(7+7)/2=7$. Midpoint is 7 (value 17). $17 \leq 17$ is true, so the search range is updated to $L=7$. $17 > 17$ is false, so the search range is updated to $L=7$.

Algoritmo de Búsqueda Binaria

```
i= 0;  
j= tamaño-1;  
do {  
    k= (i+j)/2;  
    if (v[k]<=b )  
        i=k+1;  
    if (v[k]>=b )  
        j= k-1;  
} while (i<=j);
```

include<exercise.h>

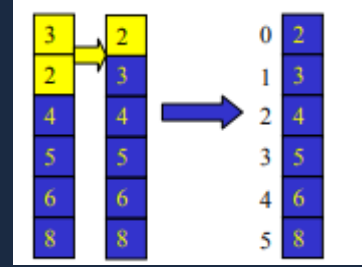
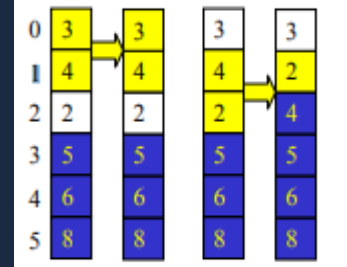
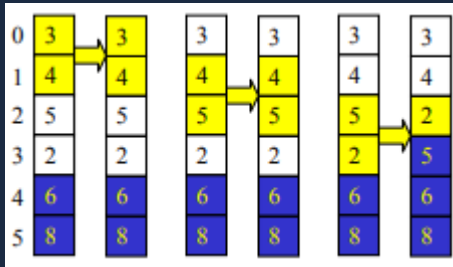
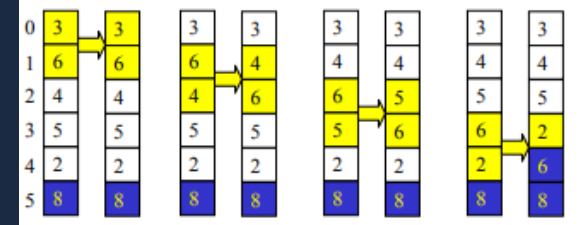
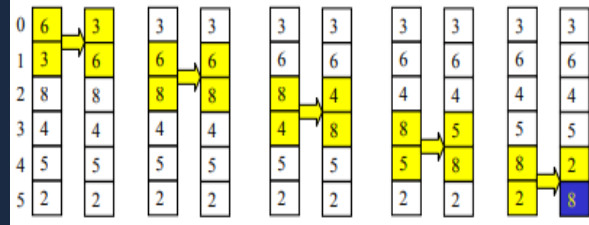
1. Escribe un programa en C que solicite al usuario ingresar un número entero n que representará el tamaño de un vector. Luego, el programa debe permitir al usuario ingresar n números enteros para llenar el vector. El programa debe pedir un número entero adicional para buscar en el vector utilizando búsqueda lineal e imprimir la posición (índice) del número si se encuentra o un mensaje indicando que el número no está en el vector.
2. Escribe un programa en C que primero pida al usuario ingresar el tamaño n de un vector de números enteros y luego los n elementos del vector de manera ordenada. Implementa la búsqueda binaria para buscar un número entero especificado por el usuario e imprime la posición del número en el vector si se encuentra o un mensaje indicando que no está presente. Asegúrate de que el vector esté ordenado para que la búsqueda binaria funcione correctamente.
3. Escribe un programa en C que solicite al usuario ingresar el tamaño n de un vector y luego los n elementos del vector. El programa debe recorrer el vector para encontrar y mostrar el valor máximo y mínimo presentes en el vector.
4. Escribe un programa en C que pida al usuario ingresar el tamaño n de un vector y luego los n elementos del vector. Después, el programa debe solicitar un número entero al usuario y contar cuántas veces aparece ese número en el vector. El resultado debe ser mostrado al usuario.
5. Escribe un programa en C que permita al usuario ingresar dos vectores: uno principal de tamaño n y otro subvector de tamaño m . El programa debe buscar la primera ocurrencia del subvector en el vector principal e imprimir la posición de inicio de esta ocurrencia, o un mensaje indicando que el subvector no se encuentra en el vector principal. Por ejemplo, si el vector principal es [1, 2, 3, 4, 2, 3, 4, 5] y el subvector es [2, 3, 4], el programa debe imprimir 1 (índice de la primera aparición de 2 en el vector principal).

Ordenamiento BubleSort (Burbuja)

Vamos comparando elementos adyacentes y empujamos los valores más livianos hacia arriba (los más pesados van quedando abajo). Idea de la burbuja que asciende, por lo liviana que es.

Algoritmo de BubleSort

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



include<exercise.h>

1. Escribe un programa en C que solicite al usuario ingresar el tamaño n de un vector y luego los n números enteros que compondrán el vector. Implementa el algoritmo de ordenamiento por burbuja para ordenar los números en orden ascendente. Finalmente, muestra el vector ordenado al usuario.
2. Escribe un programa en C que solicite al usuario ingresar el tamaño n de un vector (asegúrate de que n sea impar) y luego los n números enteros que compondrán el vector. Usa el algoritmo de ordenamiento por burbuja para ordenar el vector en orden ascendente y luego encuentra y muestra el valor mediano (el elemento en la posición central del vector ordenado).
3. Escribe un programa en C que permita al usuario ingresar un conjunto de palabras (cadenas de caracteres). Solicita primero el número de palabras n y luego las n palabras. Usa el algoritmo de ordenamiento por burbuja para ordenar las palabras en función de su longitud, de la más corta a la más larga. Muestra las palabras ordenadas.
4. Escribe un programa en C que solicite al usuario ingresar el tamaño n de un vector y luego los n números enteros que compondrán el vector. Implementa el algoritmo de ordenamiento por burbuja para ordenar el vector en orden ascendente. Después de ordenar el vector, elimina los elementos duplicados y muestra el vector resultante sin duplicados.
5. Escribe un programa en C que solicite al usuario ingresar el tamaño n de un vector y luego los n números decimales que compondrán el vector. Usa el algoritmo de ordenamiento por burbuja para ordenar los números en orden ascendente. Después de ordenar el vector, calcula y muestra el promedio de los números.