



Métodos de Ordenación

Programación I



Métodos

Existen muchos algoritmos de búsquedas diversos, cada uno intenta minimizar el tiempo o esfuerzo en encontrar el o los elementos, y varían a su vez las precondiciones o supuestos para su aplicabilidad, por ejemplo, si los datos están ordenados, o no, si hay elementos repetidos, si el elemento seguro se encuentra o no, etc.

En nuestra cursada estudiaremos dos tipos de algoritmos, las búsquedas secuenciales y las binarias.



Búsqueda secuencial

Aplicabilidad:

- Desconocimiento acerca de la organización de los datos
- Estructura solo accedida
- Visitar todas las posiciones el encuentre el elemento o se llegue al final del mismo (elemento no esta)

PROCEDURE Secuencial (V:Vector; x:TipoElemento)

VAR ind: int;

begin

ind := 0;

while ((ind < N) & (V[ind] <> x)) do

ind := ind + 1;

if (ind = N) then

write ("no encontrado")

else

write ("encontrado");

End;



Búsqueda secuencial

Consideraciones

- La expresión lógica sólo es correcta si el segundo término sólo se evalúa cuando el primero es TRUE
- Costo
 - Mejor Caso 1 comparación
 - Peor Caso N comparaciones
 - Caso Promedio $\sim N/2$ comparaciones
- Fin de Búsqueda:
 - Elemento hallado $V[ind]=x$
 - Elemento no hallado $ind=N$



Búsqueda Binaria

Aplicabilidad:

Información adicional: Cómo están organizados los datos.

Búsqueda más eficiente Datos Ordenados.

Idea Clave:

Inspeccionar un elemento de índice m elegido al azar (x elemento a buscar):

- Si $V[m]=x$ Fin Búsqueda
- Si $V[m]<x$ " $k/ k \in m$, $V[k]$ eliminados
- Si $V[m]>x$ " $k/ k \in m$, $V[k]$ eliminados



Búsqueda Binaria

```
Procedure Binaria (V:Vector; x:TipoElemento)
VAR
  izq,der,m:int; encontrado:boolean;
begin
  Izq := 1;
  Der := N
  Encontrado := false;
  while ((Izq <= Der) & (!Encontrado)) do begin
    m := izq + ((der – izq) div 2);
    If (V[m] = x) then
      Encontrado := true;
    Else begin
      if (V[m] < x) then
        izq := m + 1
      else
        der := m – 1;
    End;
  End;
End;
```



Búsqueda Binaria

Consideraciones

- Objetivo: Eliminar el mayor número de elementos en cada iteración
- Elección Optima $m := (\text{Izq} + \text{der})/2$
- Eficiencia (Peor Caso) $\text{Trunc}(\log_2 N) + 1$