

ALGORITMOS Y PROGRAMACIÓN

Clase 9 - Segunda Parte
Búsqueda

Temario

- Búsqueda
 - Búsqueda secuencial
 - Búsqueda binaria

Ordenamiento y búsqueda

- El tener una colección de datos ordenada facilita las tareas de búsqueda de elementos e inserción de nuevos datos a la colección.
- Los métodos de búsqueda tienen por objetivo recuperar información almacenada en algún dispositivo de memoria.

Búsqueda secuencial

- La búsqueda secuencial busca un elemento de una colección utilizando un valor destino llamado clave. En una búsqueda secuencial (a veces llamada búsqueda lineal), los elementos de una lista o vector se exploran (se examinan) en secuencia, uno después de otro.
- El algoritmo de búsqueda secuencial ***compara cada elemento de la lista con la clave de búsqueda*** y dado que la lista no está en un orden prefijado, es probable que el elemento a buscar pueda ser el primer elemento, el último elemento o cualquier otro.

Búsqueda secuencial

```
private static bool busquedaSecuencial(int [] datos, int elemento){  
  
    foreach(int e in datos)  
        if(elemento == e){  
            return true;  
        }  
    return false;  
}
```

O bien de esta otra forma también se codifica:

```
boolean esta=false;  
foreach(int e in datos)  
    if(elemento == e){  
        esta= true;  
        break;  
    }  
return esta;
```

Búsqueda binaria

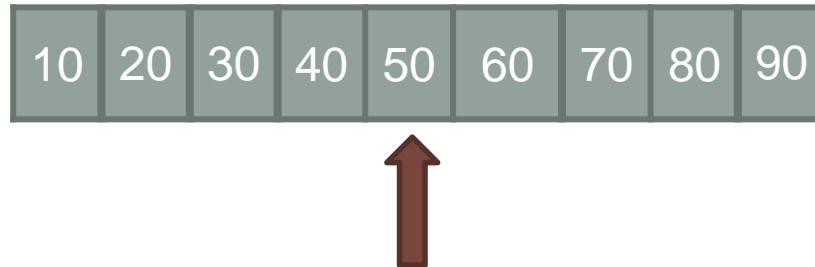
- **Si la colección en donde buscar se encuentra ordenada** entonces es posible utilizar la búsqueda binaria que permite encontrar un elemento de forma más rápida que la secuencial.
- Se sitúa la lectura en el centro de la colección y se comprueba si la clave coincide con el valor del elemento central. Si no se encuentra el valor de la clave, se sigue la búsqueda en la mitad inferior o superior del elemento central de la colección.

Búsqueda binaria

- Se desea saber si el valor 80 se encuentra en la colección
- Si *primero* es el índice más bajo a buscar y *ultimo* el más alto, el índice central se calcula como:
 - $(\text{primero} + \text{ultimo}) / 2$

Recordar que en C# los elementos de las colecciones de datos se numeran desde 0

primero = 0
ultimo=8
central
 $(0+8)/2 = 4$



La búsqueda comienza en la mitad del arreglo. Posición 4.

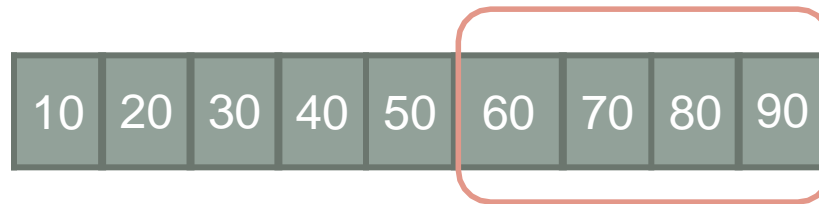
Búsqueda binaria

10	20	30	40	50	60	70	80	90
----	----	----	----	----	----	----	----	----



El 80 es más grande que el
elemento central (50).

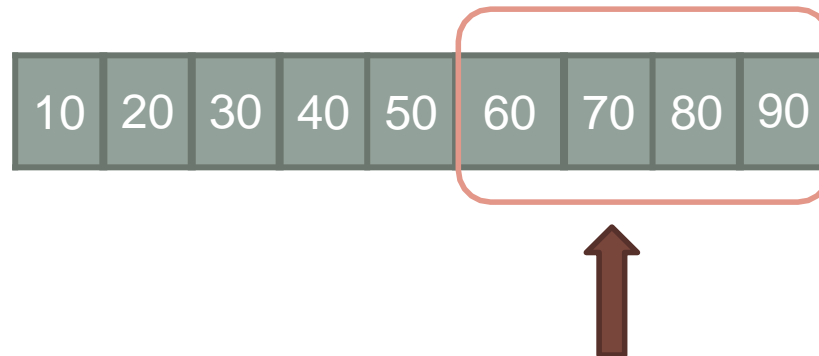
Búsqueda binaria



Se sigue la búsqueda en la
mitad superior

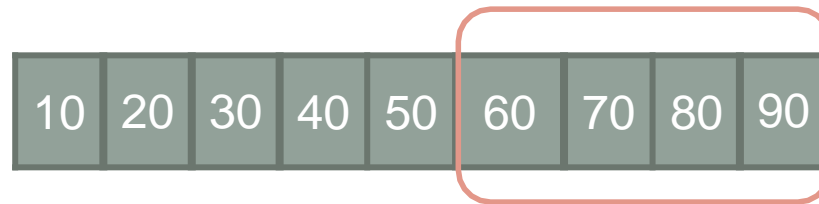
Búsqueda binaria

primero = 5
ultimo=8
central→
 $(5+8)/2 = 6$



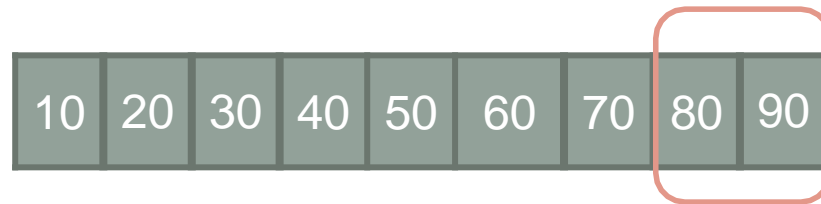
Se calcula el nuevo centro (de la subcolección a buscar).
Posición 6

Búsqueda binaria



El 80 es más grande que el elemento de la posición central(70).

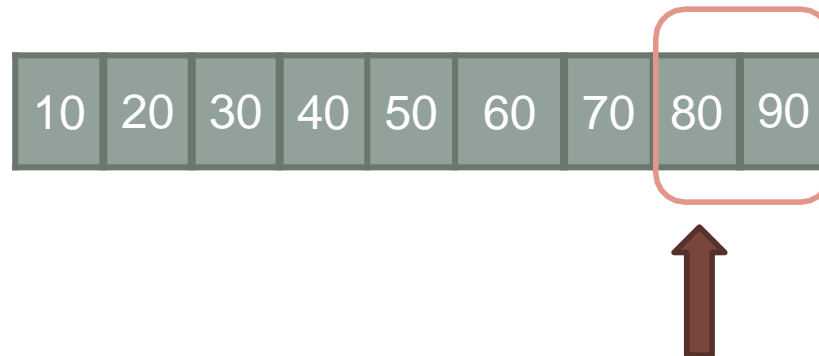
Búsqueda binaria



Se continúa la búsqueda en la mitad superior

Búsqueda binaria

primero = 7
ultimo = 8
central →
 $(7+8)/2 = 7$



Se calcula el nuevo centro (de la subcolección a buscar).

Búsqueda binaria

- Fin del algoritmo

10	20	30	40	50	60	70	80	90
----	----	----	----	----	----	----	----	----

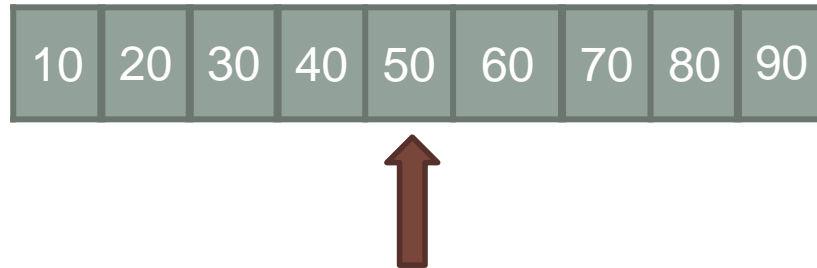


Se encontró el elemento buscado

Búsqueda binaria

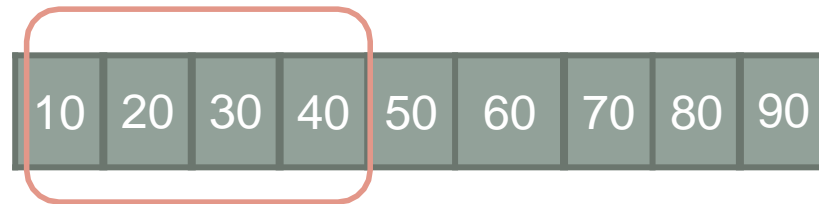
- Veamos otro ejemplo : se desea saber si el valor 34 se encuentra en la colección
- Si *primero* es el índice más bajo a buscar y *ultimo* el más alto, el índice central se calcula como:
 - $(\text{primero} + \text{ultimo}) / 2$

primero = 0
ultimo = 8
central →
 $(0+8)/2 = 4$



La búsqueda comienza en la
mitad del arreglo

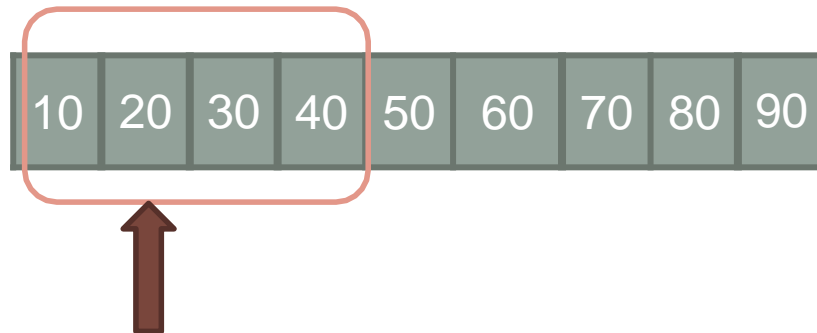
Búsqueda binaria



El valor 34 es más chico que el valor central (50), se busca en la mitad inferior

Búsqueda binaria

primero = 0
ultimo = 3
central →
 $(0+3)/2 = 1$



Se redefine primero. Luego
se calcula el nuevo centro.
Posición 1.

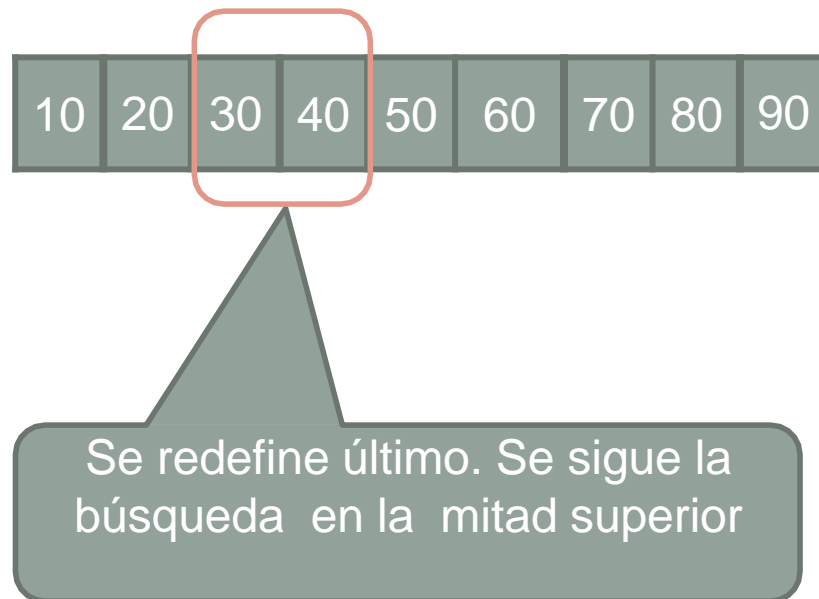
Búsqueda binaria

10	20	30	40	50	60	70	80	90
----	----	----	----	----	----	----	----	----



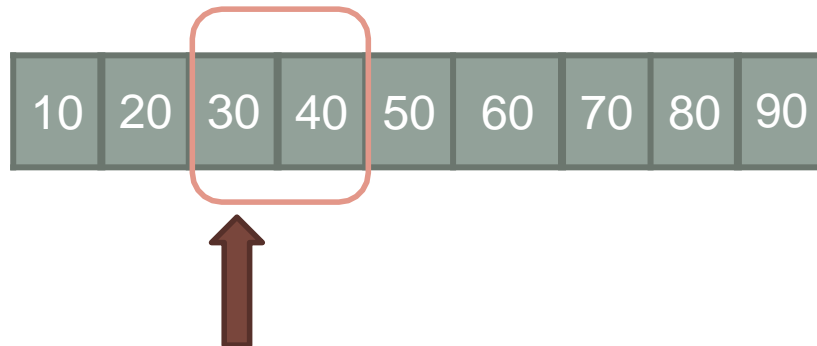
El 34 es mayor que el valor
central (20)

Búsqueda binaria



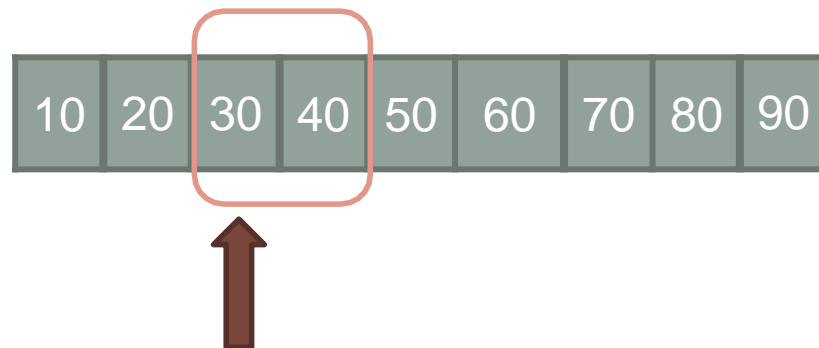
Búsqueda binaria

primero = 2
ultimo = 3
central →
 $(2+3)/2 = 2$



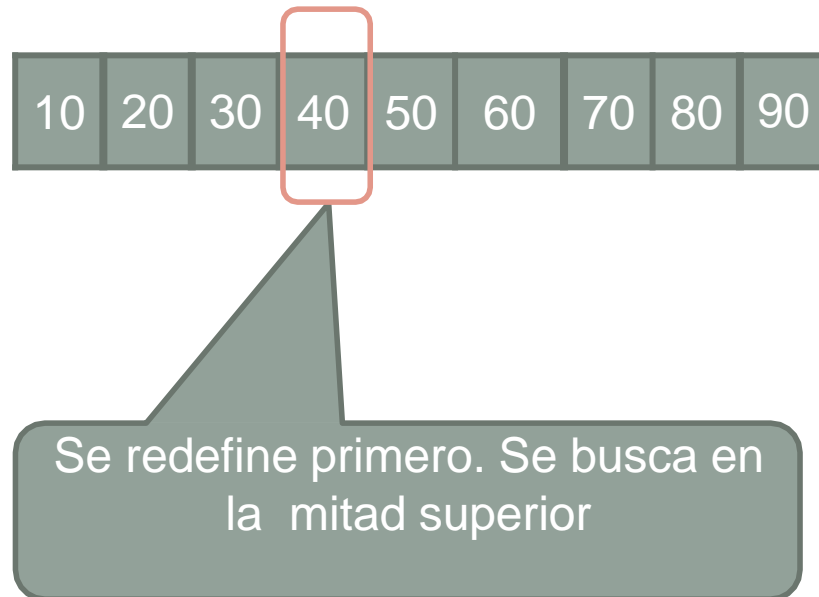
Se calcula el nuevo centro (de la subcolección a buscar)

Búsqueda binaria



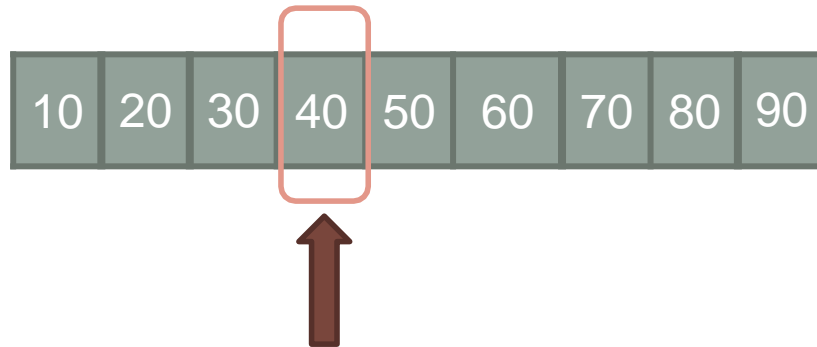
El 34 es mayor que el elemento
de la posición central(30)

Búsqueda binaria



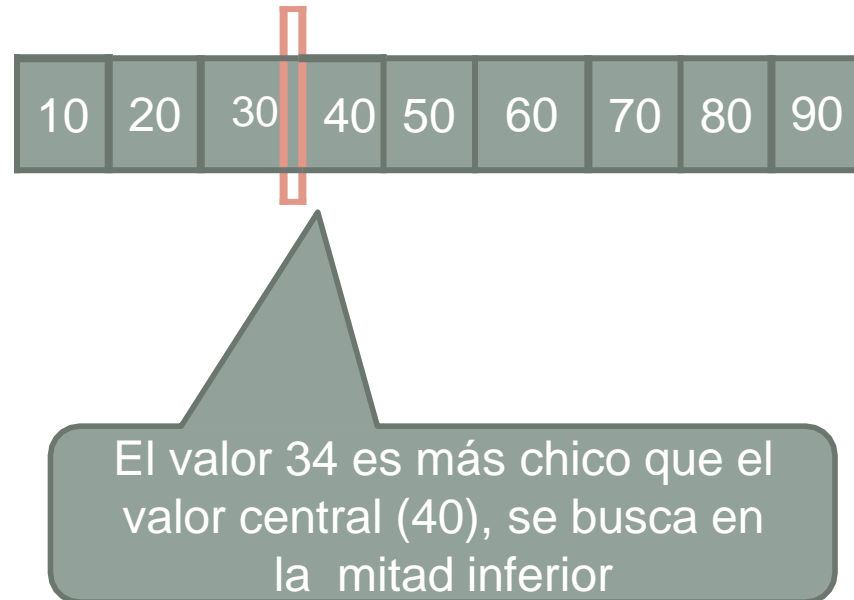
Búsqueda binaria

primero = 3
ultimo = 3
central →
 $(3+3)/2 = 3$



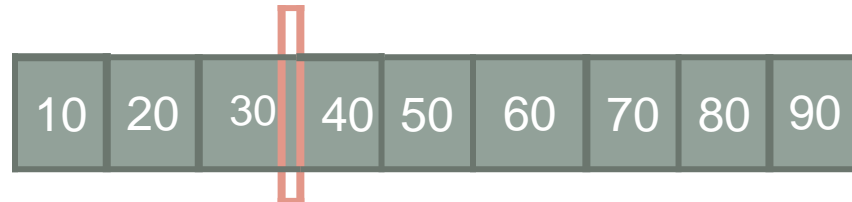
Se calcula el nuevo centro (de la subcolección a buscar)

Búsqueda binaria



Búsqueda binaria

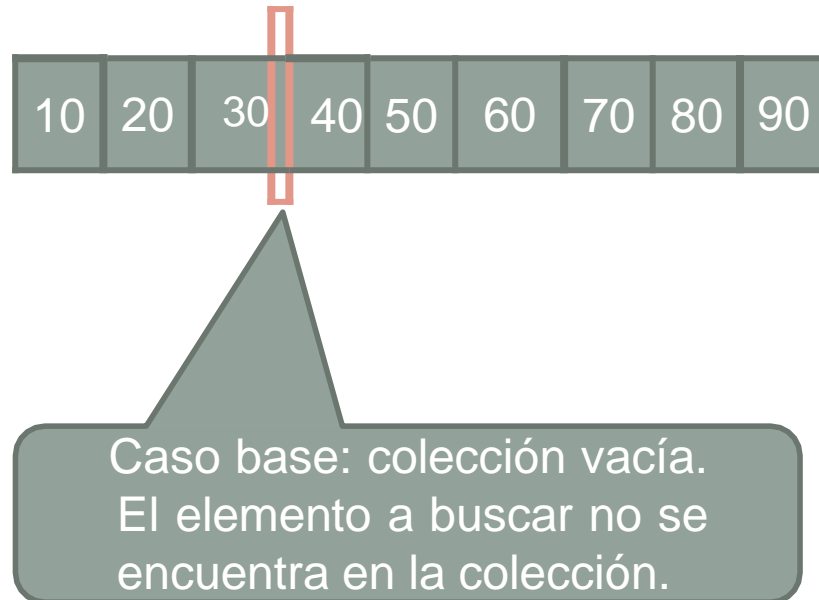
primero = 3
ultimo = 2



Se redefine último, pero los
índices se cruzan.....

Búsqueda binaria

- Fin del algoritmo



Búsqueda binaria

*/*programa iterativo*/*

```
public static bool busquedaBinaria(int [] datos, int elemento){
    int primero, ultimo;
    bool esta=false;
    primero=0;
    ultimo=datos.Length - 1;
    int central;
    while ( (primero <= ultimo) && (esta==false))
    {
        central= (primero + ultimo) / 2;
        if(datos[central] == elemento)
            esta=true;
        else
        {
            if (datos[central] >
            elemento)
                ultimo = central - 1;
            else
                primero = central + 1;
        }
    }
    return esta;
}
```

/* programa recursivo*/

```
public static bool busquedaBinaria(int [] datos, int elemento, int primero, int ultimo){  
    int central = (primero + ultimo) / 2;  
    if(datos[central] == elemento){  
        return true;  
    }  
    else{  
        if(datos[central] > elemento){  
            ultimo = central - 1;  
            if(ultimo < primero){  
                return false;  
            }  
        }  
        else{  
            primero = central + 1;  
            if(primero > ultimo){  
                return false;  
            }  
        }  
        return busquedaBinaria(datos, elemento, primero, ultimo);  
    }  
}
```

