

Tarea 1 - Análisis y Diseño de Algoritmos

Algoritmos de Búsqueda

1. Introducción

El problema que intentan resolver las búsquedas en programación es decidir si existe un elemento dado x en una lista de enteros ordenado. La **búsqueda lineal** consiste en buscar de forma secuencial el elemento en la lista dada. Esta búsqueda compara cada elemento con el buscado, hasta que lo encuentra. La **búsqueda binaria** consiste en dividir una lista (u otra estructura de datos) por su elemento medio en dos sublistas más pequeñas, y comparar el elemento que se está buscando con el del centro. Si coinciden, la búsqueda se termina. Si el elemento es menor, debe estar (si está) en la primera sublista, y si es mayor, está en la segunda. Y se repite este proceso hasta encontrar dicho elemento. A continuación, se presenta el algoritmo iterativo de la búsqueda binaria

Algorithm 1 Búsqueda Binaria Iterativa

```
1: function BUSQUEDABINARIAITERATIVA(miLista, valor)
2:   inferior = 0
3:   superior = len(miLista) - 1
4:   while inferior ≤ superior do
5:     medio =  $\frac{superior + inferior}{2}$ 
6:     if miLista[medio] = valor then
7:       return medio
8:     else
9:       if valor < miLista[medio] then
10:        superior = medio - 1
11:      else
12:        inferior = medio + 1
13:      end if
14:    end if
15:  end while
16:  return -1
17: end function
```

Ahora, se presenta el algoritmo recursivo de la búsqueda binaria

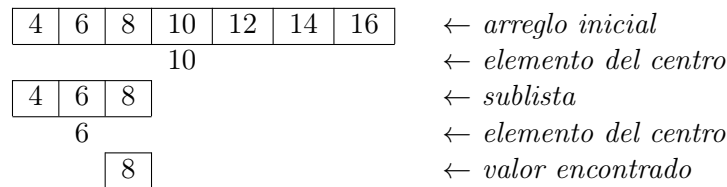
Algorithm 2 Búsqueda Binaria Recursiva

```

1: function BUSQUEDABINARIARECURSIVA(miLista, valor, primero, ultimo)
2:   if ultimo < primero then
3:     return -1
4:   else
5:     medio =  $\frac{(\text{superior} + \text{inferior})}{2}$ 
6:     if valor = miLista[medio] then
7:       return medio
8:     else
9:       if valor < miLista[medio] then
10:        return BUSQUEDABINARIARECURSIVA(miLista, valor, primero, medio - 1)
11:      else
12:        return BUSQUEDABINARIARECURSIVA(miLista, valor, medio + 1, ultimo)
13:      end if
14:    end if
15:  end if
16: end function

```

Vea el siguiente ejemplo, se desea buscar el valor 8 en una lista de números enteros ordenados:



Por otro lado, la **búsqueda ternaria (o terciaria)** utiliza la misma idea que la búsqueda binaria, salvo que en vez de dividir la lista por la mitad, la divide en tres partes, por lo que ahora se tendrían que analizar tres sublistas (en el caso iterativo), o llamar a la función con el tercio de la lista que interesa (en el caso recursivo).

2. Tarea

Lo que usted debe realizar es lo siguiente:

1. Determinar el orden de cada uno de los algoritmos: búsqueda lineal, binaria (iterativo y recursivo) y ternaria (iterativo y recursivo)
2. Desarrollar en C la búsqueda lineal, binaria (iterativo y recursivo) y ternaria (iterativo y recursivo).
3. Completar la siguiente tabla con el tiempo de ejecución de sus códigos, para una lista de **n** elementos:

n	Búsqueda				
	Lineal	Binaria (Iterativo)	Ternaria (Iterativo)	Binaria (Recursivo)	Ternaria (Recursivo)
10 ²					
10 ³					
10 ⁴					
10 ⁵					
10 ⁶					
10 ⁷					
10 ⁸					

4. Comente sobre los resultados obtenidos empíricamente y el orden de cada algoritmo obtenido teóricamente.
5. Comente sobre qué tipo de búsqueda es mejor y por qué. Fundamente claramente su respuesta.

3. Reglas del Juego

- La realización de la tarea se realiza en grupo, con un máximo de 2 integrantes.
- Se debe entregar un archivo comprimido ZIP denominado **tarea1-apellido1-apellido2.zip**, que contiene el código **busqueda.c** y el informe **busqueda.pdf**.
- El archivo comprimido debe ser entregado en la tarea **Tarea 1** que estará disponible en el Aula Virtual hasta el **Domingo 17 de Septiembre** a las **23:55 hrs**.
- Al comienzo del código, debe ir comentado los nombres de los integrantes del grupo.
- Cualquier caso de copia, se evaluará a los grupos involucrados con nota 1,0.