

Búsqueda Binaria
Corporación Minuto de Dios
Facultad de ingeniería

Presentado por: Cristian David Mora Sáenz NRC: 7487

Profesor: Segundo Fidel Puerto Garavito

Asignatura: Diseño de algoritmos

Marzo de 2020, Bogotá - Colombia

Índice

1. Introducción	2
2. Método de búsqueda binaria	3
2.1. Como utilizar el método de búsqueda binaria:	4
2.2. Tiempo de ejecución:	5
3. Conclusiones	6

1. Introducción

En este documento se explica el funcionamiento del algoritmo que corresponde al método de búsqueda binaria, este método como su nombre lo indica se utiliza para buscar elementos almacenados en una estructura de datos, pero a diferencia de métodos de búsqueda de tipo secuencial (que recorren toda la estructura para encontrar el elemento o ítem) este método de búsqueda binaria se encarga de utilizar la longitud de la estructura, para reducir la cantidad de búsquedas a la mitad y realizar el proceso más rápido, este algoritmo en especial es realizado de forma recursiva.

2. Método de búsqueda binaria

El metodo de busqueda binaria se utiliza para encontrar elementos en estructuras de datos lineales (Arrays, Listas, Pilas o colas), este método recibe como parametros dos elementos especificos (*arreglo* e *Ítem buscado*), estos parametros se utilizan para realizar las comparaciones de datos a través de la estructura.

A continuacion se muestra una imagen del algoritmos recursivo desarrollado en python:

```
def busquedaBinaria(lista, item):
    if len(lista) == 0:
        return False
    else:
        medio = len(lista)//2
        if lista[medio] == item:
            return True
        else:
            if item < int(lista[medio]):
                return busquedaBinaria(lista[:medio], item)
            else:
                return busquedaBinaria(lista[medio+1:], item)
```

Como se puede observar el método de búsqueda binaria se utiliza a partir de una serie de condicionales IF- ELSE, para registrar si el valor contenido en la casilla de la mitad de la estructura corresponde al ítem buscado, además si este no es el dato, el algoritmo utiliza como condicional el valor de esa casilla para saber si el valor es mayor o menor y realizar el mismo proceso mediante recursividad.

2.1. Como utilizar el método de búsqueda binaria:

1. Para poder ejecutar el método de búsqueda binario, primero debemos definir un arreglo o estructura de datos lineal como se muestra a continuación:

```
Arr = [1,3,5,6,7,9,10,2,4,8,12,14,16,18,19,20,11,13,15,17]
```

2. Posteriormente debemos tener en cuenta que nuestro arreglo debe ser ordenado, si no se ordena el arreglo, el método de búsqueda binaria, por defecto no encontrara el dato solicitado ya que dicho método no es secuencial y para agilizar el proceso, no recorre el arreglo si no que busca a partir de las mitades que el propio algoritmo determine necesarias. Para este ejercicio, se ordena el arreglo utilizando un ordenamiento por inserción:

```
def ordenamientoPorInsercion(unalista):  
    for indice in range(1,len(unalista)):  
        valorActual = unalista[indice]  
        posicion = indice  
        while posicion>0 and unalista[posicion-1]>valorActual:  
            unalista[posicion]=unalista[posicion-1]  
            posicion = posicion-1  
        unalista[posicion]=valorActual
```

3. Para finalizar el algoritmo, mostrara por consola el arreglo de forma No ordenada y ordenada, para que, utilizando el siguiente código, el usuario solicite alguno de los datos que visualiza en pantalla:

```
var = int(input("Cual de los elementos del arreglo desea Buscar --> "))  
  
busquedaBinaria(Arr, var)
```

4. Luego de tener un método de ordenamiento, se procede a utilizar el método de búsqueda, con el cual nos encargaremos de mostrar en pantalla el elemento buscado y la posición, que por defecto será el mismo valor encontrado. A continuación, se muestra el resultado obtenido al momento de ejecutar el algoritmo.

```
Arreglo no ordenado  
[1, 3, 5, 6, 7, 9, 10, 2, 4, 8, 12, 14, 16, 18, 19, 20, 11, 13, 15, 17]  
Arreglo Ordenado  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]  
Cual de los elementos del arreglo desea Buscar --> 14  
El número: 14 se encontro en la posición: 14
```

2.2. Tiempo de ejecución:

Podemos observar el tiempo de duración que a tenido la ejecución de nuestro algoritmo, teniendo como resultado: **0.0010004043579101562** segundos

3. Conclusiones

Se concluye que, al utilizar el método de búsqueda binaria recursivo, es un método muy eficiente, para la implementación en un algoritmo que utiliza un tipo de estructura lineal, como es el caso en este ejercicio, además, si el método es utilizado junto con un método de ordenamiento cuyo nivel de comparaciones de elementos sea reducido reflejara un tiempo de ejecución de igual magnitud.