

Búsqueda Binaria

viernes, 3 de setiembre de 2021 23:34

BÚSQUEDA BINARIA

La Búsqueda Binaria implica buscar un dato determinado en una lista. Para esto se determina un valor medio de la lista y se compara si el elemento en esa posición contiene el dato buscado, si lo contiene, entonces se devuelve el valor. En caso contrario, se dividirá la lista considerando si el dato buscado es menor o mayor con referencia al medio actual, nos quedaremos con la mitad de la lista donde debería estar el dato. Se repite el proceso de división y comparación consecutivamente hasta encontrar el dato buscado o hasta que sea imposible seguir dividiendo la lista. En caso de no encontrar el dato retornamos -1.

La búsqueda binaria mejora el tiempo de ejecución de la búsqueda lineal, en este caso tendrá un tiempo de $O(\log N)$ porque realizamos consecutivas divisiones de la lista hasta encontrar el resultado.

La limitación de este algoritmo es que la lista deberá estar ordenada de manera ascendente.

LENGUAJE NATURAL

Determinamos el medio de la lista y verificamos si esta posición contiene el dato buscado. Si lo contiene devolvemos su valor. En caso contrario determinamos si el dato buscado se encuentra en la primera mitad de la lista (menor al valor de la posición actual) o en la segunda mitad (mayor al valor de la posición actual). Nos quedamos con la lista donde debería estar el dato y repetimos el proceso de división y verificación, así consecutivamente hasta encontrar el dato o hasta que no sea posible realizar más divisiones, esto nos indica que no se encuentra un elemento con ese valor, en este caso devolvemos -1.

PRECONDICIONES

La lista puede contener o no elementos.

La lista debe estar ordenada de manera ascendente (condición excluyente).

POSTCONDICIONES

La lista no se modifica.

Los elementos de la lista no se modifican.

El método devuelve el valor buscado o -1 en caso de no existir en la lista.

PSEUDO

busquedaSecuencial (valorBuscado de tipo entero, lista de tipo arreglo de enteros) : devuelve entero

COMIENZO

 limiteINF de tipo entero <-- 0

 limiteSUP de tipo entero <-- largo de la lista

 Medio de tipo entero <-- redondeo al menor de $(\text{limiteSUP} + \text{limiteINF}) / 2$

 Continuar de tipo booleano <-- verdadero

 MIENTRAS continuar = verdadero HACER

 SI lista[medio] = valorBuscado ENTONCES

 Continuar <-- falso

 Devolver lista[medio]

 FinSi

 SI $(\text{limiteINF} = \text{medio} \text{ OR } \text{limiteSUP} = \text{medio})$ ENTONCES

 Continuar <-- falso

 FinSi

 SI $(\text{valorBuscado} < \text{lista}[\text{medio}])$ ENTONCES

 limiteSUP <-- medio

 medio <-- redondeo a menor de $(\text{limiteSUP} + \text{limiteINF}) / 2$

 SINO

 limiteINF <-- medio

JAVA
