

ALGORITMOS DE BUSQUEDA

Una de las maneras para buscar un elemento en un array, consiste en pasar por cada una de las posiciones hasta que sea encontrado. Siempre y cuando existiera dentro de esa lista de elementos almacenados. El algoritmo sería el siguiente:

N, tamaño de la tabla, n° entero. P, n° enteros.
tabla(N), array de n° enteros
elemento, n° entero, elemento a encontrar.

INICIO

Leer elemento

P = 1

Mientras (elemento \neq tabla[P]) AND P \leq N

 P = P + 1

Fin_Mientras

Si P > N

 visualizar "ELEMENTO NO ENCONTRADO"

sino

 visualizar "ELEMENTO ENCONTRADO"

Finsi

FIN

Podríamos utilizar una variable para detectar si ha sido encontrado dentro del bucle, esa variable la denominamos **swieth o centinela**, nosotros le damos un valor inicial y cuando ocurre la situación que nosotros buscamos le cambiamos el valor para detectarlo y parar el proceso. El algoritmo es similar al de arriba, pero con una variable más.

N, tamaño de la tabla, n° entero. P, n° enteros.
tabla(N), array de n° enteros
elemento, n° entero, elemento a encontrar.
sw, n° entero, valor inicial 0

INICIO

Leer elemento

P = 1

sw = 0

Mientras sw = 0 AND P \leq N

 Si tabla[P] = elemento

 visualizar "ELEMENTO ENCONTRADO"

 sw = 1

 sino

 P = P + 1

 Fin_Si

Fin_Mientras

FIN

Si la tabla con la cual trabajamos, esta ordenada podemos utilizar otros métodos más eficientes. Hablamos de **búsqueda binaria**.

Consiste en localizar un elemento con el menor número posible de preguntas. Como recordareis ya lo hicimos en el ejercicio para adivinar la clave secreta. Consistía en ir preguntado en el elemento que se encontraba en la mitad, y de esta manera desechábamos la mitad de los elementos restantes; volviendo a dividir a la mitad los restantes y volver a preguntar.

N, tamaño de la tabla, nº entero.

P, J, nº enteros.

tabla(N), array de nº enteros **ordenado**

mitad, nº entero, para seguir la búsqueda sobre la mitad superior o inferior de la tabla.

primero, ultimo, nº enteros cuyo valor inicial es 1 y N.

elemento, nº entero, elemento a encontrar.

INICIO

Leer elemento

mitad = (primero + ultimo) / 2

Mientras (tabla[mitad] <> elemento) AND (primero <= ultimo) **

 Si elemento < tabla[mitad]

 ultimo = mitad - 1

 sino

 primero = mitad + 1

 Finsi

mitad = (primero + ultimo) / 2

Fin_Mientras

Si elemento = tabla[mitad]

 visualizar "ELEMENTO ENCONTRADO"

 sino

 visualizar " EL ELEMENTO NO SE ENCUENTRA"

Fin_Si

FIN

** En Java, como el primer elemento del array toma la posición 0, debemos preguntar por (primero < ultimo)

El mismo procedimiento de búsqueda binaria podemos realizarlo de forma recursiva. Seria el que viene a continuación.

N, tamaño de la tabla, n° entero.

P, J, n° enteros.

tabla(N), array de n° enteros **ordenado**

mitad, n° entero, para seguir la búsqueda sobre la mitad superior o inferior de la tabla.

primero, ultimo, n° enteros cuyo valor inicial es 1 y N.

elemento, n° entero, elemento a encontrar.

FUNCION BUSCAR(primero, ultimo, tabla, elemento)

mitad = (primero + ultimo) / 2

Si tabla[mitad]= elemento

visualizar “ELEMENTO ENCONTRADO”

sino

Si tabla[mitad]> elemento

ultimo = mitad

sino

primero = mitad

Finsi

Si ((primero + ultimo) / 2) = mitad

visualizar “ EL ELEMENTO NO SE ENCUENTRA”

sino

BUSCAR(primero, ultimo, tabla, elemento)

Fin_Si

Fin_Si

FIN_FUNCION