

Búsqueda binaria

Búsqueda binaria

Entrada: Arreglo X con N elementos ordenados ascendentemente y un valor A

Salida: Índice en la que se encuentra A dentro de X ó -1 si no se encuentra

Búsqueda binaria

```
function busquedaBinaria(X, N, A):  
    return buscar(X, A, 0, N-1)
```

```
function buscar(X, A, i, j)  
    if i <= j:  
        k = (i+j)/2  
        if A =  $X_k$ :  
            return k  
        else if A <  $X_k$ :  
            buscar(X, A, i, k-1)  
        else  
            buscar(X, A, i+1, j)  
    else:  
        return -1
```

Búsqueda binaria

Uso del método maestro:

a (número de subproblemas): 1

b (factor de “encogimiento”): 2

d (complejidad “interna”): 0

$$T(n) = \begin{cases} O[N^d \text{Log}_b(N)] & \text{Si } a = b^d & \text{(caso 1)} \\ O(N^d) & \text{Si } a < b^d & \text{(caso 2)} \\ O[N^{\text{Log}_b(a)}] & \text{Si } a > b^d & \text{(caso 3)} \end{cases}$$

$$\rightarrow T(n) = O[N^0 * \text{Log}_2(N)] = O(\log(N))$$

Búsqueda binaria

En Java se puede usar *Arrays.binarySearch*

En Python se puede usar *bisect_left* de la librería *bisect*