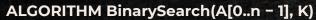
Binary

É um algoritmo eficiente para encontrar um elemento em um array ordenado, comparando o elemento do meio com o elemento de busca e repetindo a busca na metade esquerda ou direita do array, reduzindo o espaço de busca pela metade a cada iteração até encontrar o elemento desejado ou determinar que ele não está presente.



//Implements nonrecursive binary search

//Input: An array A[0..n - 1] sorted in ascending order and

// a search key K

//Output: An index of the array's element that is equal to K

// or -1 if there is no such element

I ← 0; r ← n − 1

while l ≤ r do

m ← □(I + r)/2□

if K = A[m]return m

else if $K<A[m] r \leftarrow m - 1$

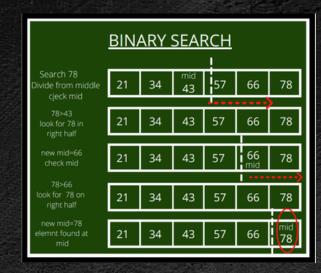
else I ← m + 1

return -1

Pior Caso

seudocódigo

$$C_{worst}(n) = \lfloor \log_2 n \rfloor + 1 = \lceil \log_2(n+1) \rceil.$$



Sequential

Algoritmo simples de força bruta que compara cada elemento de uma lista com o item de busca até encontrar uma correspondência, ou até percorrer toda a lista sem encontrar uma correspondência.

Search Algoritmos de busca

ALGORITHM SequentialSearch2(A[0..n], K)

//Implements sequential search with a search key as a sentinel

//Input: An array A of n elements and a search key K

//Output: The index of the first element in A[0..n - 1] whose

value is

// equal to K or -1 if no such element is found

A[n]

K

i

O

while A[i] = K do

i + i + 1

if i<n return i

else return -1

Seudocódigo

