Saulo Carvalho

**Instruções do projeto**

Temos o array [15, 8, 10, 25, 12, 30, 5, 20, 18, 7] e precisamos criar um código em JavaScript para encontrar o valor 20.

Para isso, temos que escolher entre realizar uma busca linear ou binária de acordo com a que for mais adequada para essa situação.

Codifique a solução mais eficiente para buscar o número 20 no array.

**Resolução**

Utilizando a Busca Linear:

function busca\_linear(array, numero\_buscado)

{

for (let i = 0; i < array.length; i++)

{

if (array[i] === numero\_buscado)

{

return i;

}

}

return -1;

}

// Programa principal

let array\_numeros = [15, 8, 10, 25, 12, 30, 5, 20, 18, 7];

let numero\_procurado = 20;

let resultado = busca\_linear(array\_numeros, numero\_procurado);

if (resultado !== -1)

{

console.log(`O número ${numero\_procurado} foi encontrado na posição ${resultado + 1} do array.`);

}else

{

console.log(`O número ${numero\_procurado} não foi encontrado no array.`);

}

Utilizando a Busca Binária:

// Função para ordenar o array

function bubble\_sort(array)

{

for (let i = 0; i < array.length; i++)

{

for (let j = 0; j < (array.length - i - 1); j++)

{

if (array[j] > array[j + 1])

{

let auxiliar = array[j];

array[j] = array[j + 1];

array[j + 1] = auxiliar;

}

}

}

return array;

}

function busca\_binaria(array, numero\_buscado)

{

let inicio = 0;

let fim = array.length - 1;

while (inicio <= fim )

{

let meio = Math.floor((inicio + fim) / 2);

if (array[meio] === numero\_buscado)

{

return meio;

}

if (array[meio] > numero\_buscado)

{

fim = meio - 1;

}else

{

inicio = meio + 1;

}

}

return -1;

}

// Programa principal

let array\_numeros = [15, 8, 10, 25, 12, 30, 5, 20, 18, 7];

let numero\_procurado = 20;

console.log(`Array original: ${array\_numeros.join(', ')}`);

let array\_ordenado = bubble\_sort(array\_numeros);

console.log(`Array ordenado: ${array\_ordenado.join(', ')}`);

let resultado = busca\_binaria(array\_ordenado, numero\_procurado);

if (resultado !== -1)

{

console.log(`\nO número ${numero\_procurado} foi encontrado na posição ${resultado + 1} do array.`);

}else

{

console.log(`\nO número ${numero\_procurado} não foi encontrado no array.`);

}

Considerações sobre os algoritmos:

* Para a resolução do problema foram utilizados os dois métodos de busca, o linear e o binário, a fim de comparar o desempenho de cada um em relação ao array apresentado;
* Foram realizados dois teste para cada método de busca. Esses são os resultados:

Busca Linear

1. Para encontrar o número 20, foram necessários 8 giros no laço;
2. Para realizar uma busca por um valor que não estava no array, foram necessários 10 giros, ou seja, a busca foi feita em todo o array.

Busca Binária

1. Para encontrar o número 20, foram necessários 2 giros no laço;
2. Para realizar uma busca por um valor que não estava no array, foram necessários 4 giros.

Em questões de desempenho, a Busca Binária leva vantagem. Porém, para utilizar este método de busca, é necessário ordenar o array, como foi feito utilizando o Algoritmo Bubble Sort (o Bubble Sort é um algoritmo de fácil implementação, no entanto, é considerado um dos piores algoritmos de ordenação);

* Levando em consideração o array apresentado no problema, acredito que a melhor solução seria a Busca Linear, pois o array possui apenas 10 valores. A Busca Binária obteve um desempenho muito bom, porém, existe a necessidade de ordenar o array, e isso exige um tempo de processamento;
* O site usado como referência para implementação e onde pode ser encontrado mais informações sobre o Algoritmo Bubble Sort, como sua funcionalidade e exemplos em códigos, pode ser acessado através deste link <https://www.geeksforgeeks.org/bubble-sort-algorithms-by-using-javascript/>.