Bubblesort 🡪 Método mais simples de ordenação 🡪 não é muito eficiente

25 57 48 37 12 92 86 33

25 57 48 37 12 92 86 33

25 48 57 37 12 92 86 33

25 48 37 57 12 92 86 33

25 48 37 12 57 92 86 33

25 48 37 12 57 92 86 33

25 48 37 12 57 86 92 33

25 48 37 12 57 86 33 92 🡪 primeira iteração

25 48 37 12 57 86 33 92

25 48 37 12 57 86 33 92

25 37 48 12 57 86 33 92

25 37 12 48 57 86 33 92

25 37 12 48 57 86 33 92

25 37 12 48 57 86 33 92

25 37 12 48 57 33 86 92 🡪 Segunda iteração

25 37 12 48 57 33 86 92

25 37 12 48 57 33 86 92

25 12 37 48 57 33 86 92

25 12 37 48 57 33 86 92

25 12 37 48 57 33 86 92

25 12 37 48 33 57 86 92 🡪 Terceira iteração

25 12 37 48 33 57 86 92

12 25 37 48 33 57 86 92

12 25 37 48 33 57 86 92

12 25 37 48 33 57 86 92

12 25 37 33 48 57 86 92 🡪 Quarta iteração

12 25 37 33 48 57 86 92

12 25 37 33 48 57 86 92

12 25 37 33 48 57 86 92

12 25 33 37 48 57 86 92 🡪 Quinta iteração

12 25 33 37 48 57 86 92

12 25 33 37 48 57 86 92

12 25 33 37 48 57 86 92 🡪 Sexta iteração

12 25 33 37 48 57 86 92

12 25 33 37 48 57 86 92 🡪 Sétima iteração 🡪 Fim do código;

Selection sort 🡪 Busca pelo menor e troca com o primeiro da fila

10 20 12 5 8 15

5 20 12 10 8 15

5 8 12 10 20 15

5 8 10 12 20 15

5 8 10 12 20 15

5 8 10 12 15 20

25 57 48 37 12 92 86 33

12 57 48 37 25 92 86 33

12 25 48 37 57 92 86 33

12 25 33 37 57 92 86 48

12 25 33 37 57 92 86 48

12 25 33 37 48 92 86 57

12 25 33 37 48 57 86 92

12 25 33 37 48 57 86 92

Insertion sort 🡪 Insere o valor na sua posição certa;

25 57 48 37 12 92 86 33

25 57 48 37 12 92 86 33

25 48 57 37 12 92 86 33

25 37 48 57 12 92 86 33

12 25 37 48 57 92 86 33

12 25 37 48 57 92 86 33

12 25 37 48 57 86 92 33

12 25 33 37 48 57 86 92 🡪 Vetor final.

**Código:**

function insertionSort(vetor){

let aux;

for(let i=1; i<vetor.length; i++){

for(let j=0; j<i;j++){

if(vetor[i] < vetor[j]){

aux = vetor[i]

vetor.splice(i,1)

vetor.splice(j,0,aux)

break

}

}

console.log(vetor)

}

}

function insertionSortNovo(vetor){

let tam = vetor.length

for(let i=1; i<tam; i++){

let atual = vetor[i]

let j

for(j=i-1; j>=0 && vetor[j] > atual;j--){

vetor[j+1] = vetor[j]

}

vetor[j+1] = atual

console.log(vetor)

}

}

function checaInteiro(num){

if(Number.isInteger(num)){

return true

}

return false

}

function checaVetor(vetor){

if(vetor.length < 50){

return true;

}

else{

return false;

}

}

function inserirElemento(vetor){

qtdeNumeros = Number(prompt("Digite a quantidade de números que serão inseridos: "))

do{

if(checaInteiro(qtdeNumeros) && qtdeNumeros>0){

while(qtdeNumeros+vetor.length>50 || !checaInteiro(qtdeNumeros) || qtdeNumeros<=0){

max = 50 - vetor.length

qtdeNumeros = Number(prompt("Número digitado ultrapassa limite do vetor, o maximo de números que podem ser inseridos é de: " + max + "\nDigite uma quantidade valida!"))

}

for(let i=0;i<qtdeNumeros; i++){

num = Number(prompt("Digite algum valor inteiro qualquer: \nPara parar o codigo, digite 1.1"))

if(checaInteiro(num)){

vetor.push(num)

}

else if(num == 1.1){

break

}

else{

while(checaInteiro(num) == false){

console.log("Valor digitado não é um número inteiro!")

num = Number(prompt("Digite algum valor inteiro qualquer: "))

}

vetor.push(num)

}

}

break

}

else{

qtdeNumeros = Number(prompt("Número digitado inválido, insira outro valor: "))

}

}while((!checaInteiro(qtdeNumeros) || qtdeNumeros <=0 ))

}

function main(){

let vetor = [25,57,48,37,12,92,86,33]

let entrada = 1;

while(entrada != 0){

entrada = Number(prompt("Escolha uma das opções a seguir: \n1-Inserir Elemento;\n2-Insertion Sort;\n0-Finalizar o programa;"))

switch(entrada){

case 1:

if(checaVetor(vetor)){

inserirElemento(vetor);

}

else{

console.log("Limite máximo do vetor alcançado\nRemova algum elemento do vetor primeiro!")

}

console.log(vetor)

break

case 2:

insertionSortNovo(vetor);

console.log(vetor)

break;

case 0:

break;

default:

alert("Opção Digitada inválida!")

entrada = Number(prompt("Escolha uma das opções a seguir: \n1-Inserir Elemento;\n2-Bubblesort; \n0-Finalizar o programa;"))

break;

}

}

}