

## Algoritmos de Ordenamiento

Metodo	Tiempo (ms)
Bubble Sort	184.966796875
Selection Sort	68.343017578125
Insertion Sort	38.985107421875
Merge Sort	9.7529296875
Quick Sort	13.471923828125

Conclusiones: El algoritmo Merge Sort

- Bubble Sort

```
function generarNumerosAleatorios(cantidad, min = 0, max = 10000) {
  let numerosAleatorios = []; // [] representa el array
  for (let i = 0; i < cantidad; i++) {
    let numero = Math.floor(Math.random() * (max - min + 1)) + min;
    numerosAleatorios.push(numero); //inserta los numeros al array
  }
  return numerosAleatorios; // genera el array con los numeros generados
}

function bubbleSort(array) {
  let n = array.length;
  for (let i = 0; i < n - 1; i++) {
    for (let j = 0; j < n - 1 - i; j++) {
      if (array[j] > array[j + 1]) {
        let temp = array[j];
        array[j] = array[j + 1];
        array[j + 1] = temp;
      }
    }
  }
  return array;
}
```

```

let numeros = generarNumerosAleatorios(10000); //cantidad de numeros a
mostrar

console.log(numeros);

console.time("Bubble Sort");
let numerosOrdenados = bubbleSort(numeros);
console.timeEnd("Bubble Sort");

console.log(numerosOrdenados);

```

- Selection Sort

```

function generarNumerosAleatorios(cantidad, min = 0, max = 10000) {
  let numerosAleatorios = []; // [] representa el array
  for (let i = 0; i < cantidad; i++) {
    let numero = Math.floor(Math.random() * (max - min + 1)) + min;
    numerosAleatorios.push(numero); //inserta los numeros al array
  }
  return numerosAleatorios; // genera el array con los numeros generados
}

function selectionSort(array) {
  let n = array.length;
  for (let i = 0; i < n - 1; i++) {
    let minIndex = i;
    for (let j = i + 1; j < n; j++) {
      if (array[j] < array[minIndex]) {
        minIndex = j;
      }
    }
    // Intercambiar el elemento mínimo encontrado con el primer
    elemento
    if (minIndex !== i) {
      let temp = array[i];
      array[i] = array[minIndex];
      array[minIndex] = temp;
    }
  }
}

```

```

    }
    return array;
}

let numeros = generarNumerosAleatorios(10000); //cantidad de numeros a
mostrar

console.log(numeros);

console.time("Selection Sort");
let numerosOrdenados = selectionSort(numeros);
console.timeEnd("Selection Sort");

console.log(numerosOrdenados);

```

- Insertion Sort

```

function generarNumerosAleatorios(cantidad, min = 0, max = 10000) {
    let numerosAleatorios = []; // [] representa el array
    for (let i = 0; i < cantidad; i++) {
        let numero = Math.floor(Math.random() * (max - min + 1)) + min;
        numerosAleatorios.push(numero); //inserta los numeros al array
    }
    return numerosAleatorios; // genera el array con los numeros generados
}

function insertionSort(array) {
    let n = array.length;
    for (let i = 1; i < n; i++) {
        let key = array[i];
        let j = i - 1;
        while (j >= 0 && array[j] > key) {
            array[j + 1] = array[j];
            j = j - 1;
        }
        array[j + 1] = key;
    }
    return array;
}

```

```

}

let numeros = generarNumerosAleatorios(10000); //cantidad de numeros a
mostrar

console.log(numeros);

console.time("Insertion Sort");
let numerosOrdenados = insertionSort(numeros);
console.timeEnd("Insertion Sort");

console.log(numerosOrdenados);

```

- Merge Sort

```

function generarNumerosAleatorios(cantidad, min = 0, max = 10000) {
  let numerosAleatorios = []; // [] representa el array
  for (let i = 0; i < cantidad; i++) {
    let numero = Math.floor(Math.random() * (max - min + 1)) + min;
    numerosAleatorios.push(numero); //inserta los numeros al array
  }
  return numerosAleatorios; // genera el array con los numeros generados
}

function mergeSort(array) {
  if (array.length <= 1) {
    return array;
  }

  const mid = Math.floor(array.length / 2);
  const left = array.slice(0, mid);
  const right = array.slice(mid);

  return merge(mergeSort(left), mergeSort(right));
}

function merge(left, right) {
  let resultArray = [], leftIndex = 0, rightIndex = 0;

```

```

    while (leftIndex < left.length && rightIndex < right.length) {
      if (left[leftIndex] < right[rightIndex]) {
        resultArray.push(left[leftIndex]);
        leftIndex++;
      } else {
        resultArray.push(right[rightIndex]);
        rightIndex++;
      }
    }

    return resultArray
      .concat(left.slice(leftIndex))
      .concat(right.slice(rightIndex));
  }
}

let numeros = generarNumerosAleatorios(10000); //cantidad de numeros a
mostrar

console.log(numeros);

console.time("Merge Sort");
let numerosOrdenados = mergeSort(numeros);
console.timeEnd("Merge Sort");

console.log(numerosOrdenados);

```

- Quick Sort

```

function generarNumerosAleatorios(cantidad, min = 0, max = 10000) {
  let numerosAleatorios = []; // [] representa el array
  for (let i = 0; i < cantidad; i++) {
    let numero = Math.floor(Math.random() * (max - min + 1)) + min;
    numerosAleatorios.push(numero); //inserta los numeros al array
  }
  return numerosAleatorios; // genera el array con los numeros generados
}

```

```
function quickSort(array) {
  if (array.length <= 1) {
    return array;
  }

  const pivot = array[array.length - 1];
  const left = [];
  const right = [];

  for (let i = 0; i < array.length - 1; i++) {
    if (array[i] < pivot) {
      left.push(array[i]);
    } else {
      right.push(array[i]);
    }
  }

  return [...quickSort(left), pivot, ...quickSort(right)];
}

let numeros = generarNumerosAleatorios(10000); //cantidad de numeros a
mostrar

console.log(numeros);

console.time("Quick Sort");
let numerosOrdenados = quickSort(numeros);
console.timeEnd("Quick Sort");

console.log(numerosOrdenados);
```