

JS NUGGETS

# Quick Sort

# in Javascript

# WHAT IS Quick Sort?

**Quick sort** algorithm chooses a random pivot (in this case we start from the last item) and sorts items smaller than it to the left and items bigger than it to the right till all items are in sorted order.

## Reason to choose quick sort

- ✓ Used for sorting large datasets
- ✓ Better than insertion sorting and merge sorting algorithms.

## When not to choose

- ✗ If your app requires the stability of data
- ✗ If array is nearly sorted list Avoid!, cause it's gonna loop through the list of items anyways

# How it works?

**1** Considering random element of an array as a pivot element.

**2** Compare and split the array elements using the value of pivot element.

**3** Repeating the algorithm to the new sub-array elements which we got from step 2.

**4** Concating all sorted values.

Best Case

$O(N \cdot \log(N))$

Worst Case

$O(N^2)$

# Quick sort Implementation:

```
const quickSort = (originalArray) => {
  if (originalArray.length <= 1) {
    return originalArray;
  }
  const leftArr = [], rightArr = [];
  const pivot = originalArray.pop();
  for (const element of originalArray) {
    // using pivot value start comparing
    element <= pivot ?
      leftArr.push(element) :
      rightArr.push(element);
  }
  // until sorting occurs
  return [...quickSort(leftArr), pivot,
    ...quickSort(rightArr)];
}

const myArray = [9, 0, 2, 19, 7, -2, 6, 1];
quickSort(myArray); // [-2, 0, 1, 2, 6, 7, 9, 19]
```