Quicksort Cheatsheet

Overview

- Quicksort is a divide-and-conquer sorting algorithm.
- It works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot.
- The sub-arrays are then sorted recursively.

Algorithm

```
def quicksort(arr, start, end):
    if start < end:
        # Partition the array
        pivot_index = partition(arr, start, end)

        # Sort the left sub-array
        quicksort(arr, start, pivot_index - 1)

        # Sort the right sub-array
        quicksort(arr, pivot_index + 1, end)

def partition(arr, start, end):
    # Select the pivot element
    pivot = arr[end]

    # Initialize the pivot index
    pivot_index = start

    # Partition the array
    for i in range(start, end):
        if arr[i] < pivot:
            arr[i], arr[pivot_index] = arr[pivot_index], arr[i]
            pivot_index += 1

    # Move the pivot element to its final position
    arr[pivot_index], arr[end] = arr[end], arr[pivot_index]

    # Return the pivot index
    return pivot_index
```

Time Complexity

- Worst-case performance: O(n^2)
- Best-case performance: O(n log n)
- Average-case performance: O(n log n)

Resources

- Quicksort Wikipedia

- GeeksforGeeks: Quicksort
- Visualgo: Quicksort