

Lab 6b: Quicksort

QuickSort is a "Divide and Conquer" algorithm. It picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of QuickSort that pick the pivot in different ways, including:

1. Always picking first element as pivot.
2. Always picking last element as pivot.
3. Always picking the middle element as pivot.
4. Picking a random element as pivot.
5. Picking a median (usually from 3 elements) as pivot.

The key process in QuickSort is `partition()`. The goal of the `partition()` function is, given an array and an element `x` of array as pivot, put `x` at its correct position in sorted array and put all smaller elements (smaller than `x`) before `x`, and put all greater elements (greater than `x`) after `x`. All this should be done in linear time.

The following link is helpful - Comparison Sorting Visualizations: <https://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html>

Your goal for this lab is to write code that implements the QuickSort sorting algorithm, and observe the behavior using two different pivot selection techniques: 1) Always selecting the first element and 2) selecting the median element from the first, middle and last elements.

To compare the behavior across runs, you will need to incorporate a mechanism to count the comparisons that occur during the sorting operation.

As in the previous sorting lab, you should include your test cases within the same module as the QuickSort functions, and record the results. You will submit your code (`quicksort.py`) and a pdf of the following table, with your observations recorded.

	Number of QuickSort Comparisons	Number of QuickSort Comparisons
Starting List	pivot = first	pivot = median of 3
Ordered, ascending		
n = 100		
n = 200		
n = 400		
n = 800		
Random		
n = 100 (average 10 runs)		
n = 200 (average 10 runs)		
n = 400 (average 10 runs)		
n = 800 (average 10 runs)		
Observed Big O() behavior, ordered with pivot = first :		
Observed Big O() behavior, ordered with pivot = median of 3 :		
Observed Big O() behavior, random with pivot = first :		
Observed Big O() behavior, random with pivot = median of 3 :		
For random list, observation regarding using first vs. median of 3 :		