

Quick sort algorithms

My choice Quicksort Algorithm: Median of three method

Quicksort Algorithm :

Step 1: If the number of elements in the array is 0 or 1
return the array.

Step 2: Excluding the pivot, divide the array into two partitions by calling the partition function.

The first partition contains elements that are \leq to the pivot

The second partition contains elements that are \geq to the pivot

Step 3: Repeat the steps 1 and 2 for both the partitions, i.e. $\text{quicksort}(\text{Partition1}) + \text{pivot} + \text{quicksort}(\text{Partition})$

```
def quickSort(array,l,r):  
    if l<r:                //stop when the number of elements in the array is 0 or 1  
        q=partition(array,left,right)  
        quickSort(array,l,q-1)  
        quickSort(array,q+1,r)
```

Median of three method:

Step 1: Consider the first, last and middle element of the array.

Step 2: Swap the elements if necessary to get $A[\text{left}] < A[\text{middle}] < A[\text{last}]$ i.e $A[\text{middle}]$ is the median of those three elements.

Step 3: Set $A[\text{centre}]$ element as the pivot and place it at the last index

Partitioning algorithm:

Step 1: Select an element of the array as the pivot based on the algorithm used(median of the three) and set $i=0$ and $j = 2\text{nd last index of the array}$.

Step 2: while $i < j$

Increment i till $\text{array}[i] \geq \text{pivot}$ is found

Decrement j till $\text{array}[j] \leq \text{pivot}$ is found

Swap elements at i and j

Step 3: swap element at i with pivot to get the smaller numbers to the left and larger numbers right side of the pivot

