quick sort

quick sort is an efficient, recursive sorting algorithm that employs a divide & conquire startegy. the array into smaller sub array sorts those sub-array and then combine them

Array: [3,6,8,10,1,2,1]

step 1) Choose the last element (1) as the pivot

step 2) partition the array around the pivot

Resulting in [1,6,8,10,3,2,1] where all element to the left of 1 are less than it

step 3) The pivot (1) is now it's current position

step 4) Recursively apply quick sort to the left sub array (already sorted) & the right sub array
[6,8,10,3,2,1]

- Advantage

Generally faster in practise composed to other o(nlogn) algorithm like merge sort. and help sort particular for large datasets.

12/11/24
7:34 PM

- disadvantage
  Recursive depth can lead to stack over flow for large array if the recursion is not optimised

* Space complexity
  quick sort is in place meaning it requires a small constant amount of additional storage space ((o logn)? for the recursive call stack)

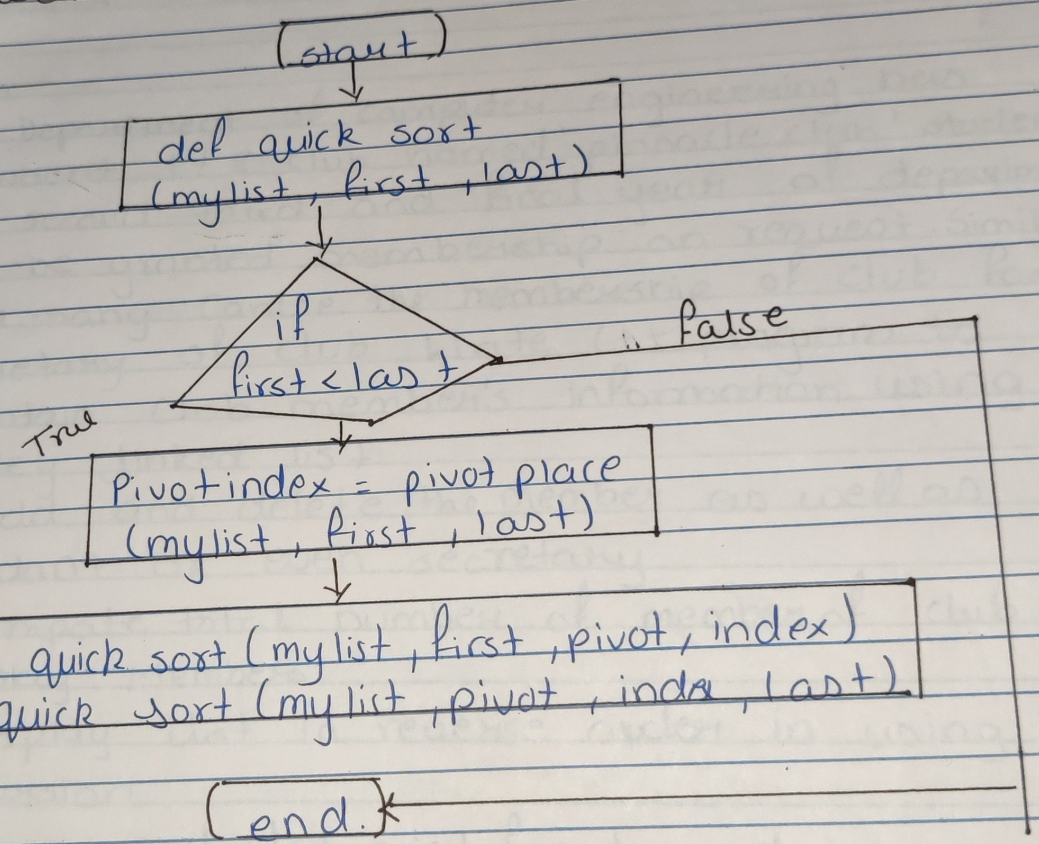
$$f(n) = O(\log n)$$

* Algorithm
step1) start
step2) choose a pivot : select an element from the array as the pivot. The choice of pivot can very (first element, last element random element or median)
step3) partition the Array : Rearrange the array around the pivot. after partitioning all element smaller than the pivot point will be it's left and all element greater than pivot will be on right.
step4) Recursively calls Recursively apply the same process to the two partitioned sub array
step5) Base Case: The recursion steps when there is only one element left in the sub.

12/11/24
7:34 PM

step 6) end.

* flowchart



* conclusion

By this way, we can perform sorting array of floating of point number in ascending order using quick sort.