# SNS COLLEGE OF ENGINEE

**Kurumbapalayam(Po), Coimbatore – 641 107**
**Accredited by NAAC-UGC with 'A' Grade**
**Approved by AICTE, Recognized by UGC & Affiliated to Anna Univer**

## Department of Artificial Intelligence an

## Course Name: 23ITB201 Data structures a

II Year / III semester

Unit III – Sorting, searching and ha

Topic: Quick Sort

a fast-sorting algorithm used to sort a list of elements. The qui

empts to separate the list of elements into two parts and then so

ly. That means it use divide and conquer strategy. In quick sor

e list is performed based on the element called pivot. Here piv

e of the elements in the list.

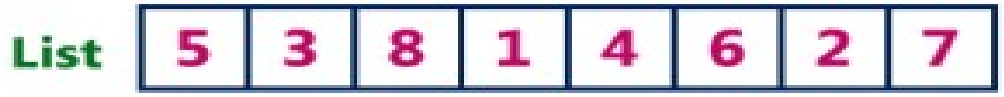
ided into two partitions such that "all elements to the left of pi

he pivot and all elements to the right of pivot are greater than

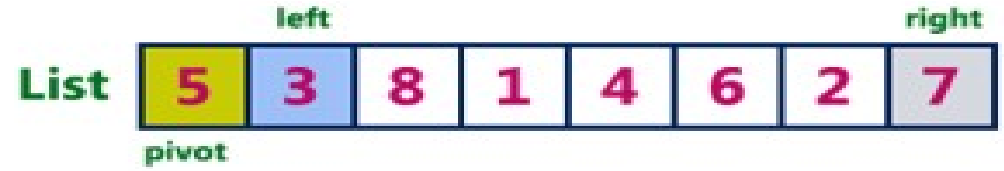
```
t[10],int first,int last){
;


;




t[i] <= list[pivot] && i < last)

t[j] > list[pivot])


 = list[i];
[i] = list[j];
[j] = temp;



ivot];
 list[j];
p;
t,first,j-1);
```

# Quick sort:

nsider the following unsorted list of elements...




efine pivot, left & right. Set pivot = 0, left = 1 & right = 7. Here '7' indicates 'size-1'.



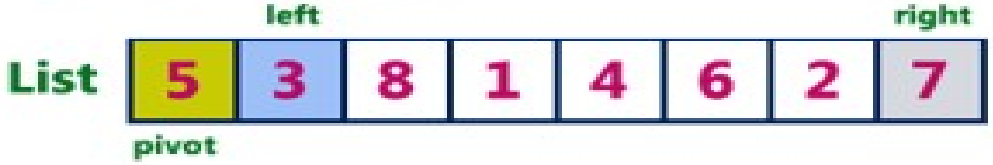

mpare List[left] with List[pivot]. If **List[left]** is greater than **List[pivot]** then stop left otherwise move left the next.

mpare List[right] with List[pivot]. If List[right] is smaller than List[pivot] then stop right otherwise move ht to the previous.

peat the same until **left>=right**.

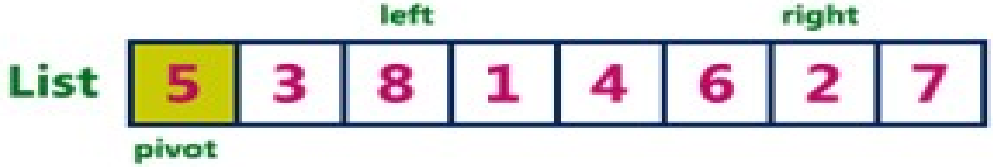
oth left & right are stoped but left<right then swap List[left] with List[right] and countinue the process.

eft>=right then swap List[pivot] with List[right].

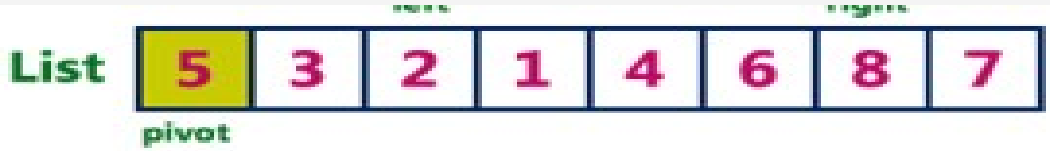



mpare List[left]<List[pivot] as it is true increment left by one and repeat the same, left will stop at 8.

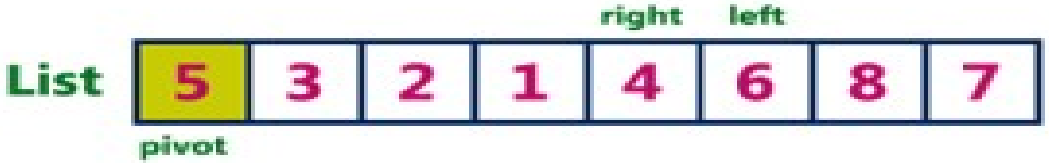
mpare List[right]>List[pivot] as it is true decrement right by one and repeat the same, right will stop at 2.



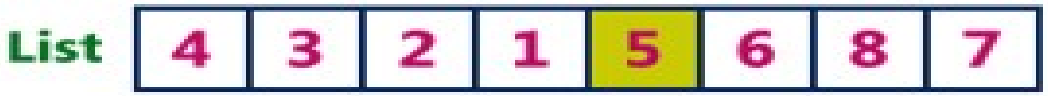
# Quick sort:




pare List[left]<List[pivot] as it is true increment left by one and repeat the same, left will stop at 6.
pare List[right]>List[pivot] as it is true decrement right by one and repeat the same, right will stop at




left & right both are stoped and left is greater than right so we need to swap List[pivot] and List[righ




we can observe that all the numbers to the left side of 5 are smaller and right side are greater. That
s 5 is placed in its correct position.
at the same process on the left sublist and right sublist to the number 5.


e left sublist as there are no smaller number than the pivot left will keep on moving to the next and s
t number. As the List[right] is smaller, right stops at same position. Now left and right both are equa
wap pivot with right.

# Quick sort:

ight so we swap pivot with right. (6 is swap by itself).


same recursively on both left and right sublists until all the numbers are sorted.

rted list will be as follows...


## the Quick Sort Algorithm

rted list with **'n'** number of elements, we need to make **((n-1)+(n-2)+(n-3)+.....+1**
in the worst case. If the list is already sorted, then it requires **'n'** number of compar