**QUICK SORT – n log n**

* Swapping by reference
* Find a pivot point in a given array take it as reference point then do operate on array (adjust array)
* All elements in left of pivot point must be smaller than pivot point…and all elements on right of pivot point must be bigger than pivot point
* Pivot point jiske right ke elements bade hone chaiye aur left wale sare chote originally
* End point = int max , start point = start point

**CODE-**

Int Quicksort(arr , start , end); {

if(start == end) return arr;

Int mid = partition(arr , start , end);

Quicksort(arr,start,mid); //left subpart

Quicksort(arr,mid+1,end); //right subpart

};

**Now creating partition function—**

* Assume start as pivot point
* Bigger elememts to end and smaller elements to start

Int Partition(arr , start , end){

Int pivot = start;

Int I = start + 1;

Int j = end;

While(I < j) {

While(arr[i] < arr[pivot]) i++;

While(arr[j] > arr[pivot]) j--;

If(I < j ) swap(arr[i] , arr[j]);

};

Swap(arr[p] , arr[j]);

Return j;

};