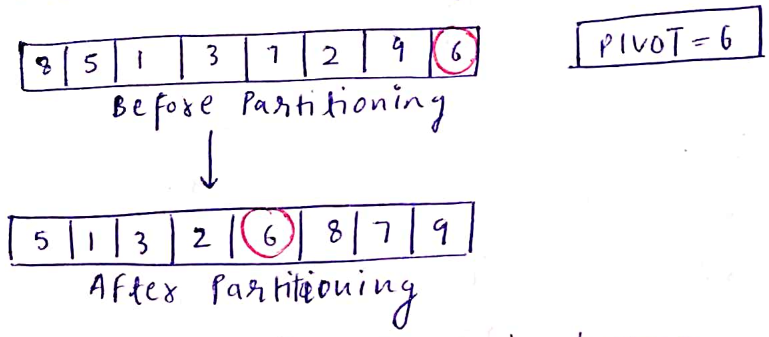
③ QUICK SORT

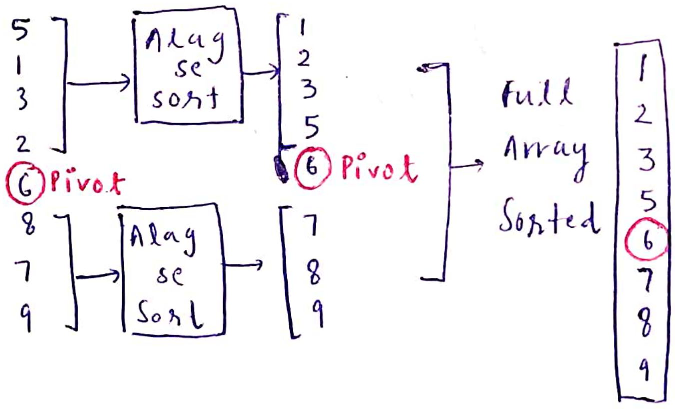
Issme hume ek array mila hoga!



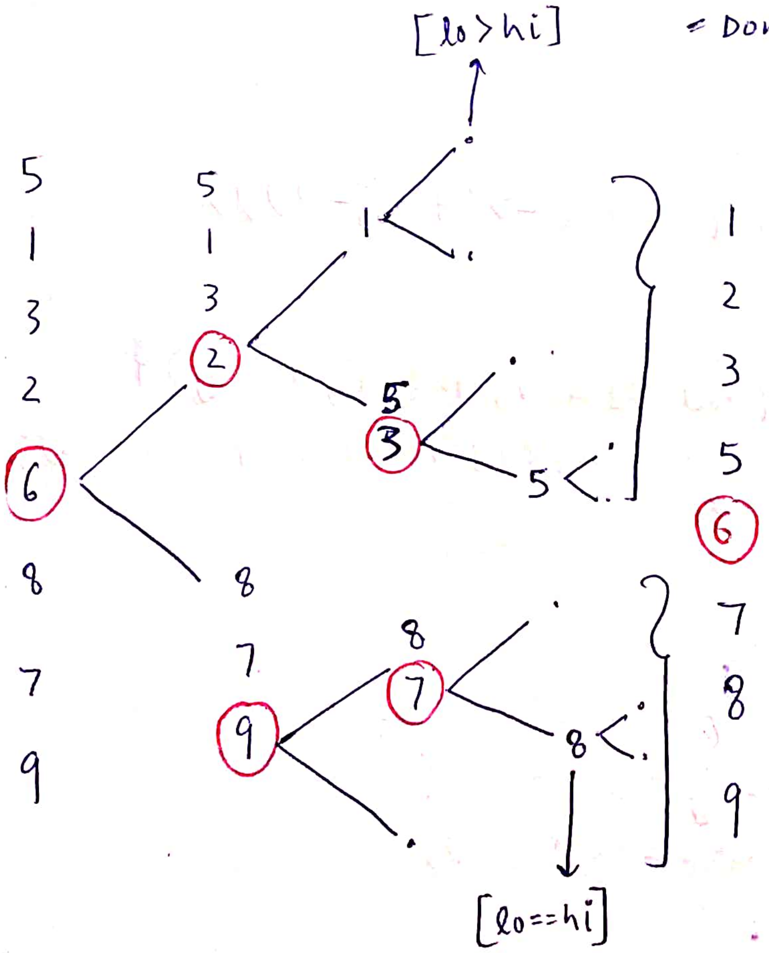
JIST

- = Partitioning kriyey
- = Chotay logo ko alag se sort honey bejo
- = Badey logo ko alag se
- = Agar choty aur badey log sort hojayengy aur middle me PIVOT hoga toh pura ARRAY hi sort hojayega!

Iss tym sabse imp baat jo hume pakadni hai vo hai ki ek element (PIVOT) sort hogaya hai!



- Last element ko PIVOT maan ke partitioning kro
- = choty elements PIVOT ke left side aur bade elements PIVOT ke right side
- Recursively left half sort hone ko boliye — {0 se pivot-1 tak}
  aur right half sort hone ko boliye! — {pivot+1 se end tak}
- = Dono half sort = pura array SORT



Base Case

```
if ( lo >= hi){
    return;
}
```

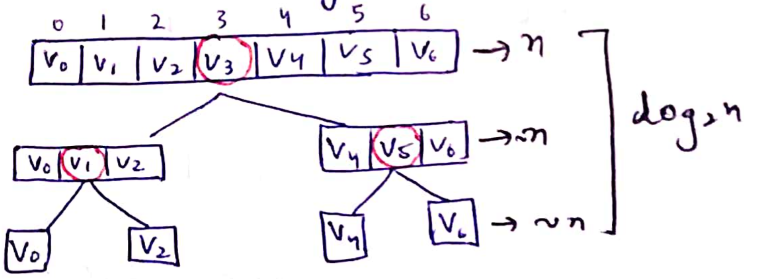
OR

```
if ( lo > hi) {
    return;
}
```

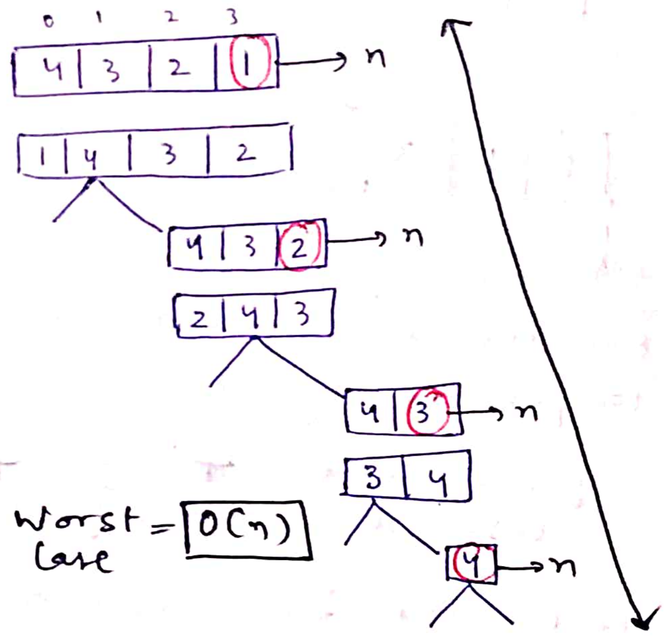
```
public static void quickSort(int[] arr, int lo, int hi) {
    if ( lo >= hi) {
        return;
    }
    int pivot = arr[hi];
    int pivotindex = partition (arr, pivot, lo, hi);
    quickSort (arr, lo, pivotindex - 1);
    quickSort (arr, pivotindex + 1, hi);

}
public static int partition (int[] arr, int pivot, int lo,
                                  int hi) {
 System.out.println( "pivot -> " + pivot);
  int  i = lo , j = lo;
  while ( i <= hi) {
      if (arr[i] <= pivot) {
          swap (arr, i, j);
           i++;
           j++;
      } else { i++; }

  }
  System.out.println( "pivot index -> " + (j-1));
  return (j-1);
}
public static void swap(int[] arr, int i, int j) {
System.out.println("Swapping" + arr[i] + "and" + arr[j]);
  int temp = arr[i];
   arr[i] = arr[j];
   arr[j] = temp;

}
public static void print (int[] arr) {
  for ( int i = 0; i < arr.length; i++) {
        System.out.print (arr[i] + " ");

   }
  System.out.println ();
}
```

```
public static void main (String [] args){
 Scanner s = new Scanner (System.in);
 int n = s.nextInt();
 int [] arr = new int [n];
 for (int i=0; i<n; i++){
     arr [i] = s.nextInt();
  }
 quickSort (arr, 0, arr.length-1);
 print (arr);
}
```

Time Complexity



∴ Best Case ⇒ $\boxed{O(n \log n)}$



Worst Case = $\boxed{O(n)}$

Space Complexity

O(1)