

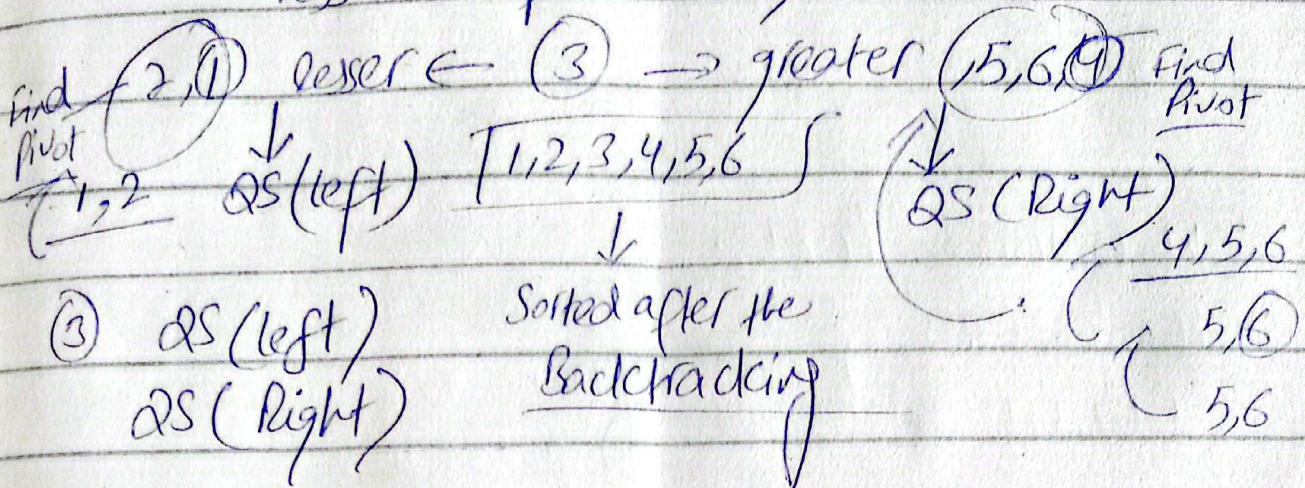
Quick Sort Algorithm:-

arr = {5, 2, 6, 4, 1, 3}

Pivot & Partition \rightarrow last Element

- ① Find Pivot (sorting is done around the pivot element)
- ② Partition on the base of pivot.

less $<$ pivot $>$ greater



Quicksort

```
void quickSort(arr, st, end) {
```

```
    if (st < end) {
```

```
        Pividx = partition(arr, st, end)
```

```
        QS(arr, st, Pividx - 1)  $\rightarrow$  left
```

```
        QS(arr, Pividx + 1, end)  $\rightarrow$  Right
```

Partition

```
int partition(arr, st, end) {
```

```
    idx = st - 1, pivot = arr[end]
```

```
    for (j = st; j < end; j++) {
```

```
        if (arr[j] < pivot) {
```

```
            idx++
```

```
            swap(arr[j], arr[idx])
```

idx = else current position

idx = st - 1

j = st to end - 1

pivot = arr[end]


```

    }
    idx++
    swap(arr[end], arr[idx])
    return idx;
}

```

$T.C = O(n \log n) \rightarrow$ Average/Practical
 worstcase = $T.C = O(n^2)$
 \downarrow pivot = smallest, largest

$S.C = O(1)$