Quick Sort

QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot.

The key process in quickSort is partition(). Target of partitions is, given an array and an element x of array as pivot, put x at its correct position in sorted array and put all smaller elements (smaller than x) before x, and put all greater elements (greater than x) after x. All this should be done in linear time.

Pseudocode for Partition

Partition(A,lb,ub)

{

pivot= A[lb];

start= lb;

end=ub;

while(start<end)

{

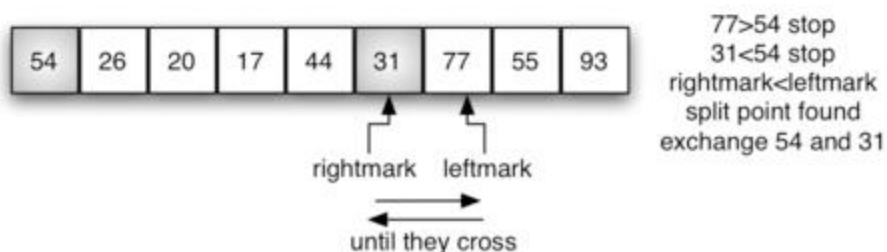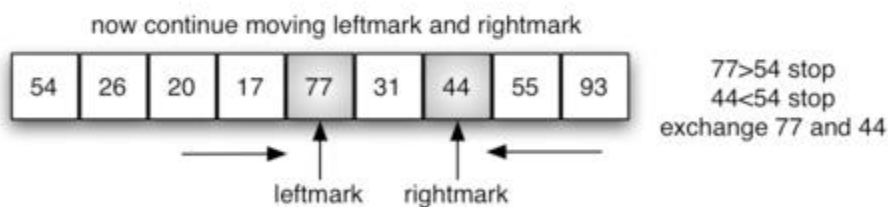while(A[start]<=pivot)
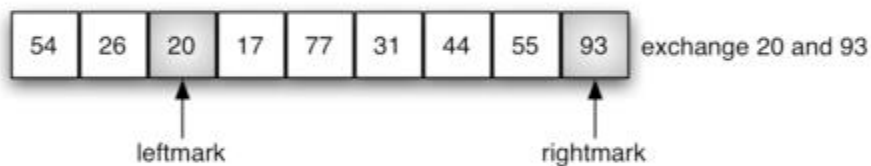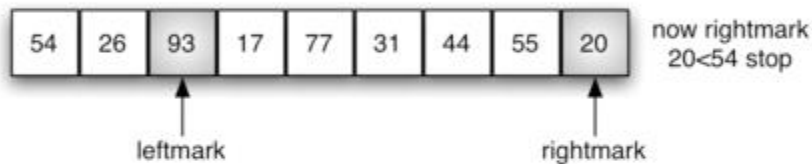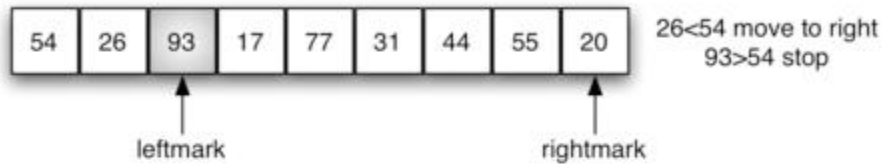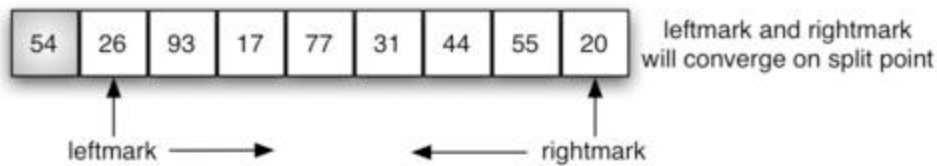
{

start++;

}

While(A[end]>pivot)

{

end--;

```
}

if(start<end)

{

Swap(A[start], A[end]);

}

}

Swap(A[lb], A[end]);

return end;

}
```

Pseudo code for Quick sort

```
Quicksort(A, lb, ub)

{

if(lb<ub)

{

loc=Partition(A,lb,ub);

Quicksort(A, lb, loc-1);

Quicksort(A, loc+1, ub);

}

}
```

How Quick Sort Works

| 54 | 26 | 93 | 17 | 77 | 31 | 44 | 55 | 20 |

54 will be the first pivot value

| 54 | 26 | 93 | 17 | 77 | 31 | 44 | 55 | 20 |

leftmark ⟶   ⟵ rightmark

leftmark and rightmark will converge on split point

| 54 | 26 | 93 | 17 | 77 | 31 | 44 | 55 | 20 |

leftmark   rightmark

26<54 move to right
93>54 stop

| 54 | 26 | 93 | 17 | 77 | 31 | 44 | 55 | 20 |

leftmark   rightmark

now rightmark
20<54 stop

| 54 | 26 | 20 | 17 | 77 | 31 | 44 | 55 | 93 |

leftmark   rightmark

exchange 20 and 93

now continue moving leftmark and rightmark

| 54 | 26 | 20 | 17 | 77 | 31 | 44 | 55 | 93 |

leftmark   rightmark

77>54 stop
44<54 stop
exchange 77 and 44

| 54 | 26 | 20 | 17 | 44 | 31 | 77 | 55 | 93 |

rightmark   leftmark

until they cross

77>54 stop
31<54 stop
rightmark<leftmark
split point found
exchange 54 and 31

| 31 | 26 | 20 | 17 | 44 | 54 | 77 | 55 | 93 | 54 is in place

<54

>54

| 31 | 26 | 20 | 17 | 44 |

quicksort left half

| 77 | 55 | 93 |

quicksort right half