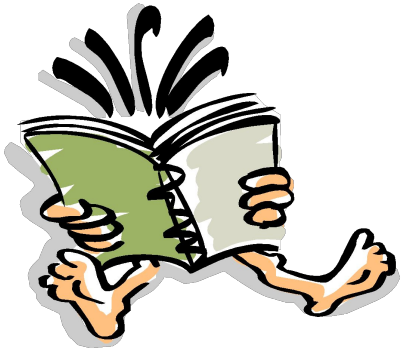
QUICK SORT

Ref Book: Cormen



Nirvruti Patel ,IT Department,UVPCE

Quick Sort

- Based On Divide and Conquer Technique
- Two Phases :
- **Partition Phase**

  Divide the input into two parts (may not be equal size)
- **Conquer Phase**

  Sort each half recursively

Quick Sort

- **Partition Phase**
- Partition the input array

1. Choose a Pivot element
2. Find the position for the pivot such that
   - All elements to the left are less
   - All elements to the right are greater

Quick Sort

- **Conquer Phase**
- Recursively sort each half

Quick Sort- Implementation

```
Algorithm QUICKSORT(A, low, high)
 if ( low <  high) {
      pivot= PARTITION(A, low, high);
      QUICKSORT(A, low, pivot-1);
      QUICKSORT(A, pivot+1,high);
}
```

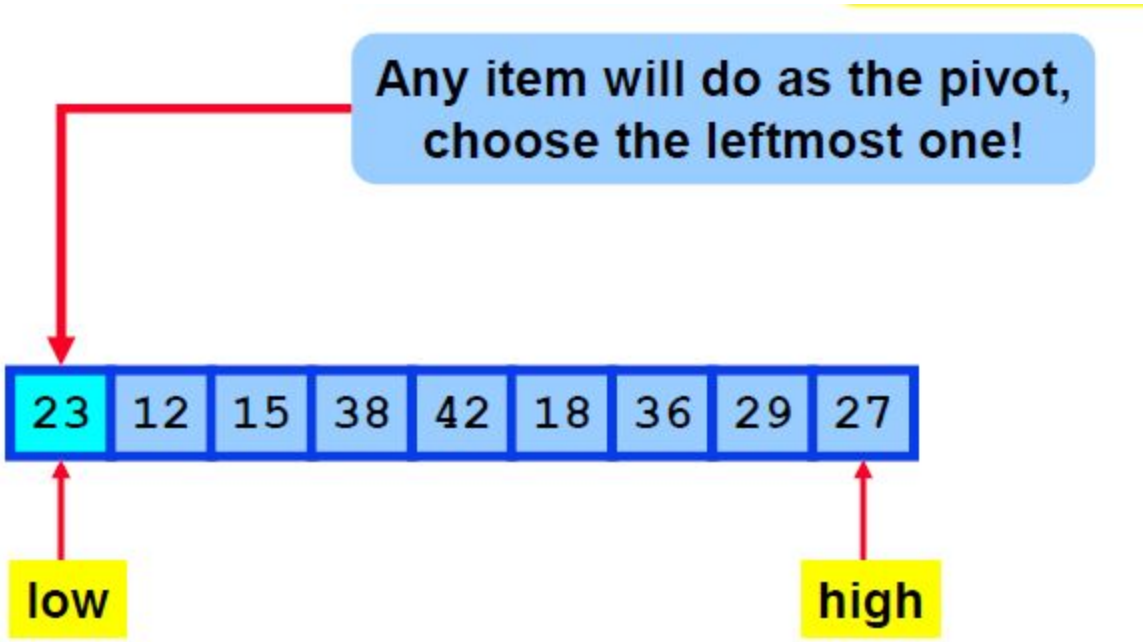
Divide

Conquer

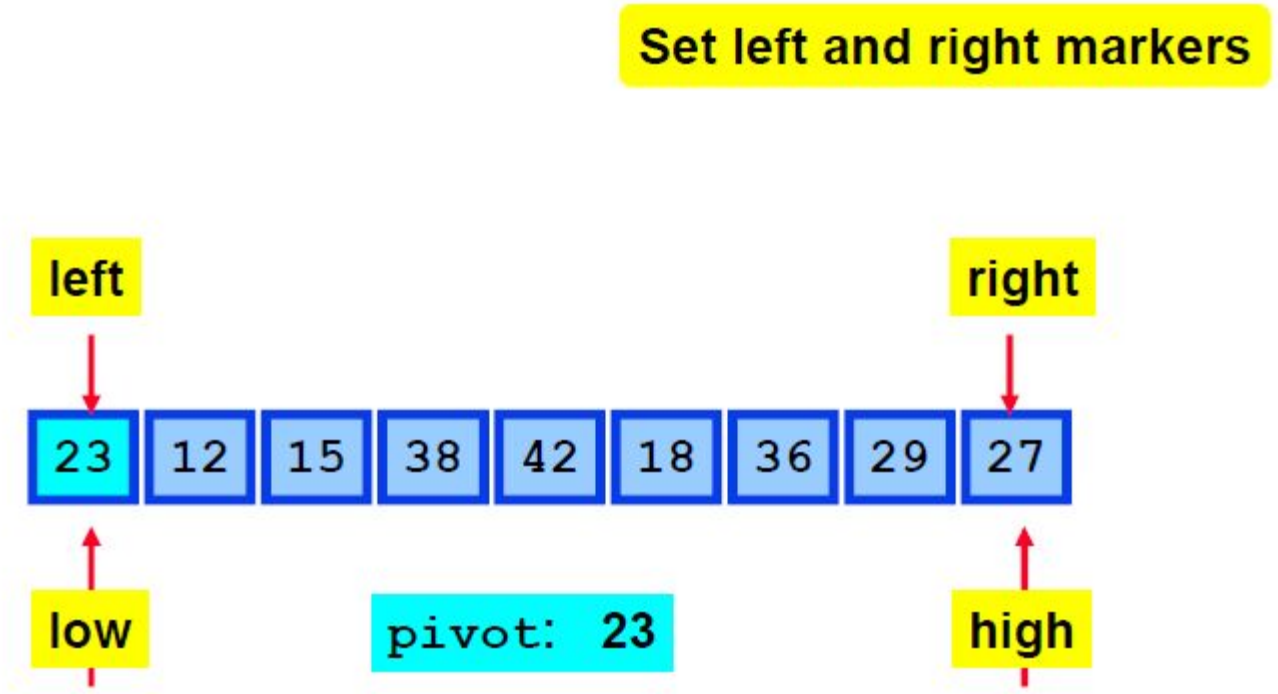
Quick Sort

```
Algorithm PARTITION(A , low, high) {
      pivot = A[low]
      left = low
      right = high
      while ( left < right ) {
            /* Move left while item < pivot */
            while( A[left] <= pivot  && left < right ) left++;
            /* Move right while item > pivot */
            while( A[right] > pivot ) right--;
            if ( left < right )
                  SWAP(A[left],A[right]);
      }
      /* right is final position for the pivot */
      SWAP(A[right], A[low]);  //   Pivot means A[low]
      return right;
}
```

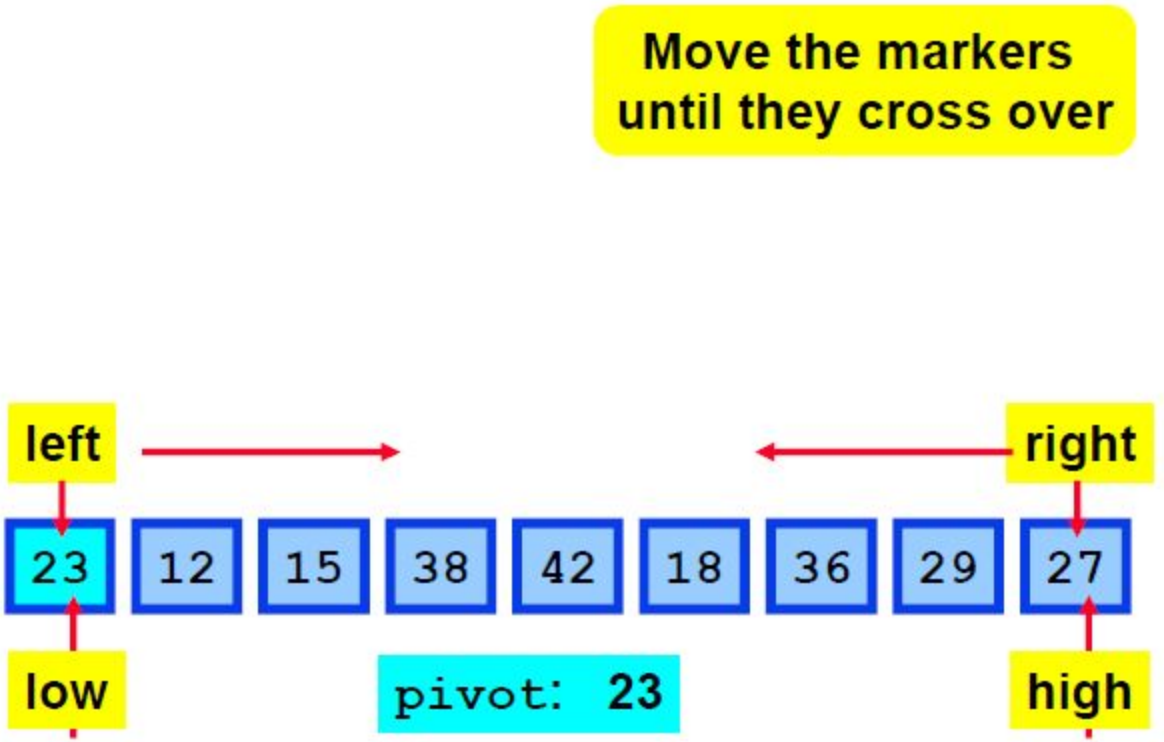
Quick Sort


Any item will do as the pivot,
choose the leftmost one!
23
12
15
38
42
18
36
29
27
low
high

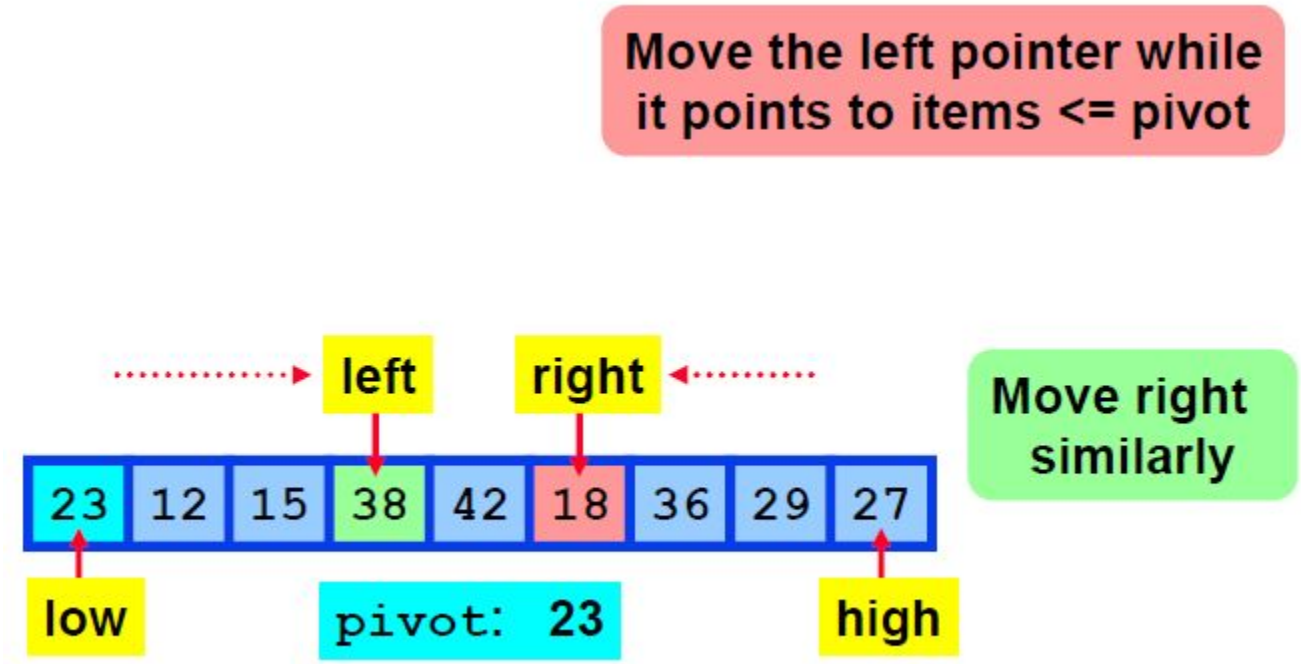
Quick Sort


Set left and right markers
left
right
23
12
15
38
42
18
36
29
27
low
pivot: 23
high

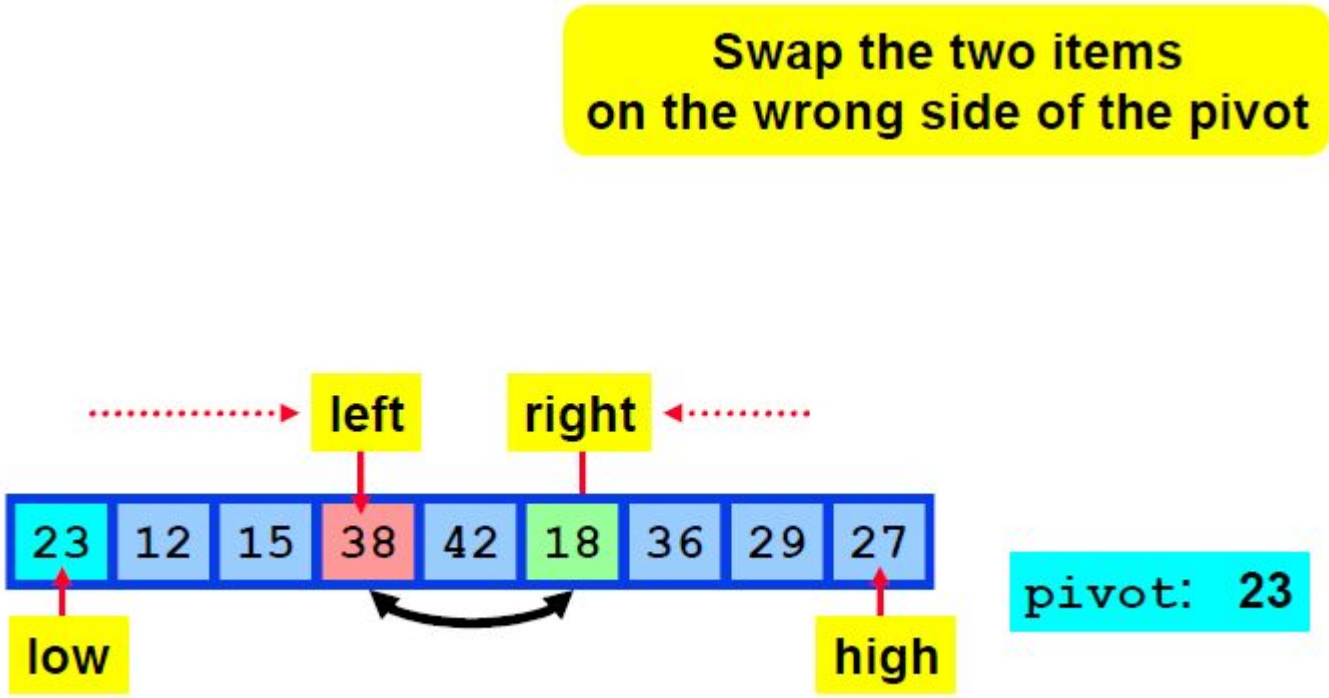
Quick Sort


Move the markers
until they cross over
left
right
23
12
15
38
42
18
36
29
27
low
pivot: 23
high

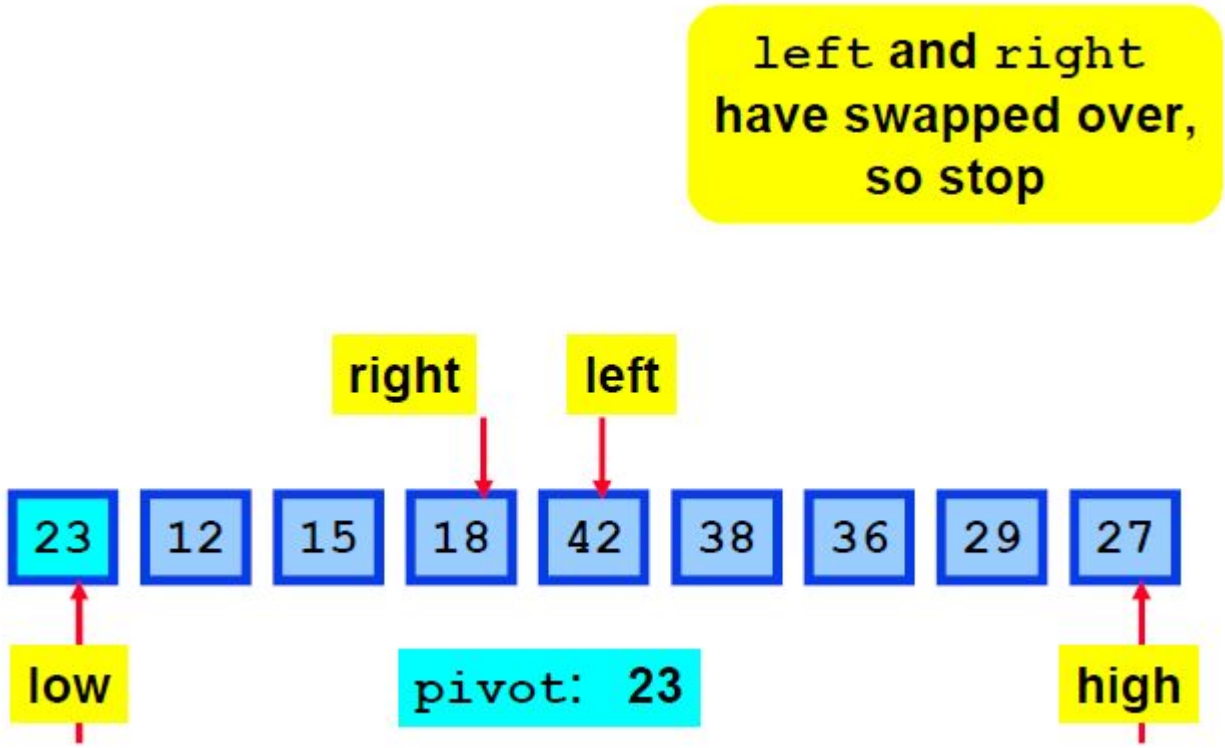
Quick Sort


Move the left pointer while it points to items <= pivot
left
right
Move right similarly
23
12
15
38
42
18
36
29
27
low
pivot: 23
high

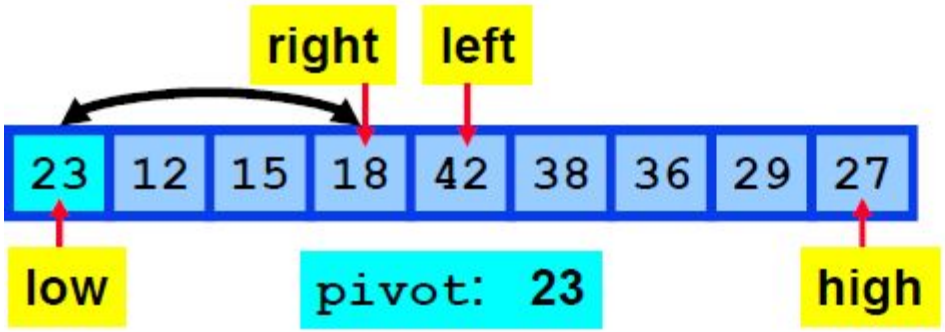
Quick Sort


Swap the two items
on the wrong side of the pivot
left
right
23
12
15
38
42
18
36
29
27
low
high
pivot: 23

Quick Sort

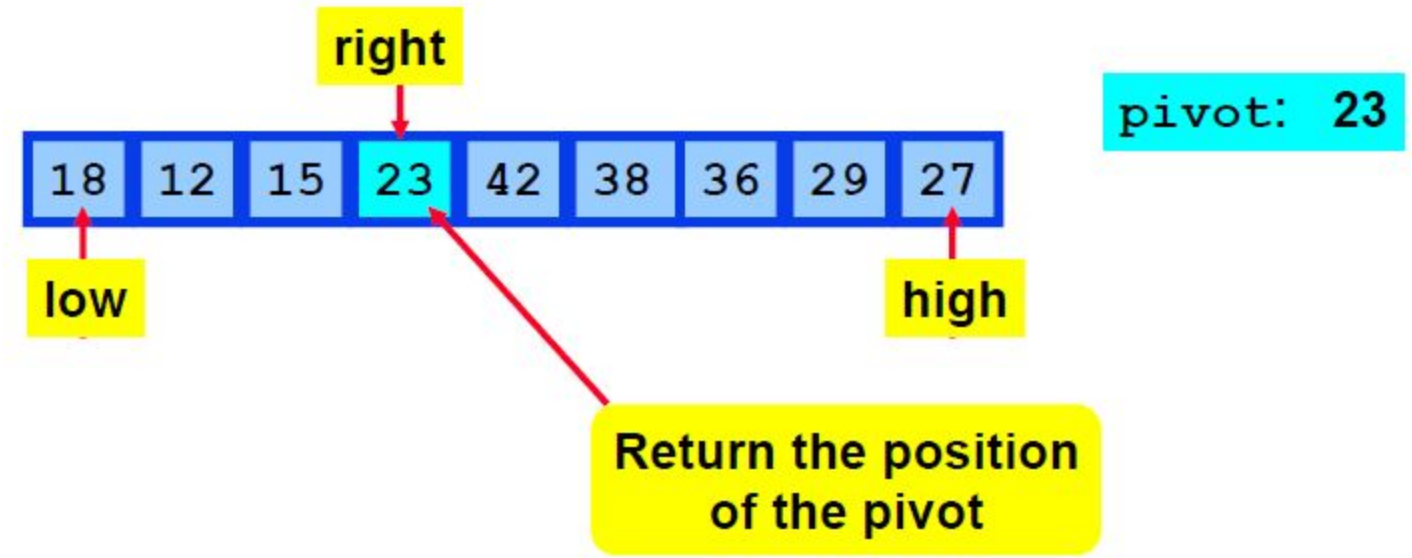

left and right have swapped over, so stop
right
left
23
12
15
18
42
38
36
29
27
low
pivot: 23
high

Quick Sort

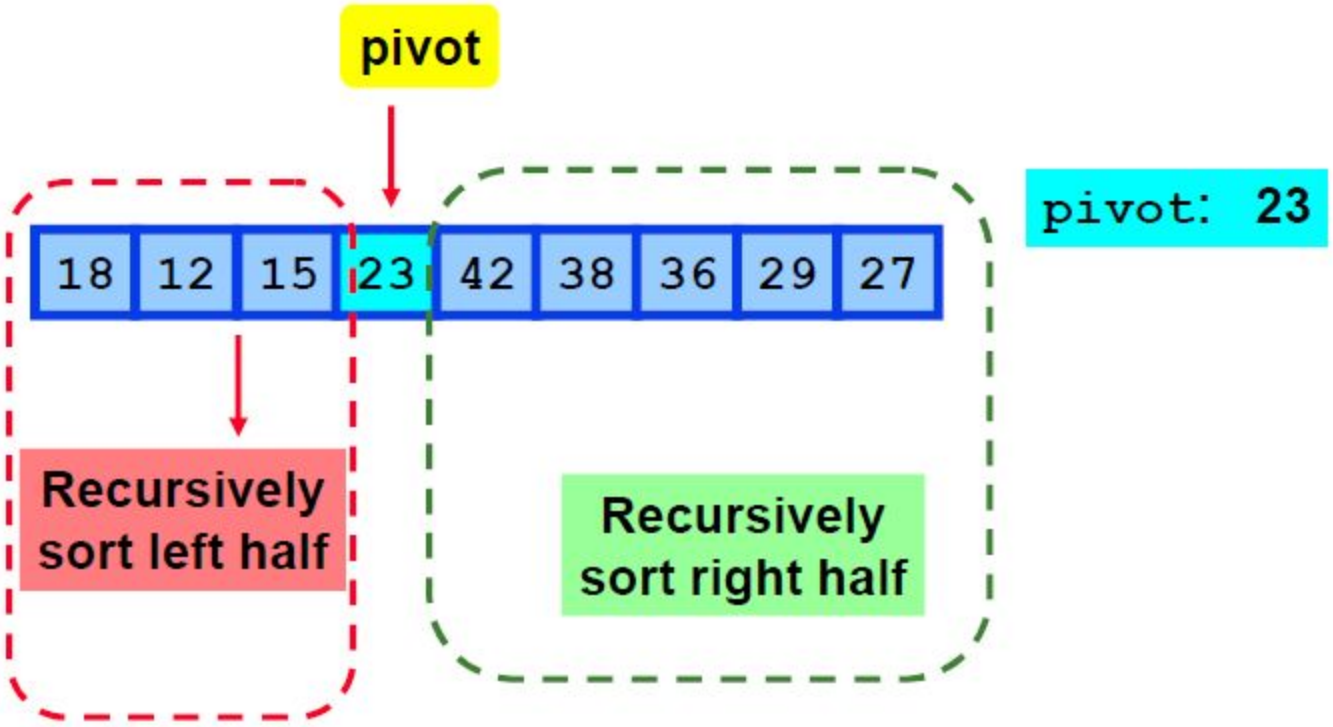


Finally, swap the pivot and `right`

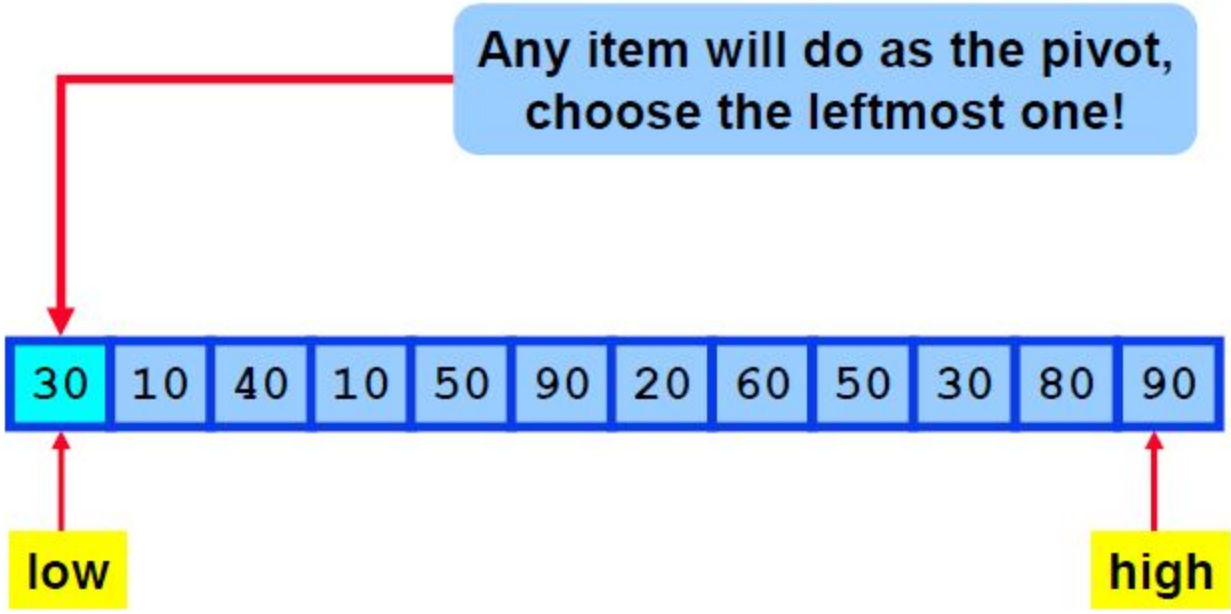
Quick Sort


right
pivot: 23
18
12
15
23
42
38
36
29
27
low
high
Return the position of the pivot

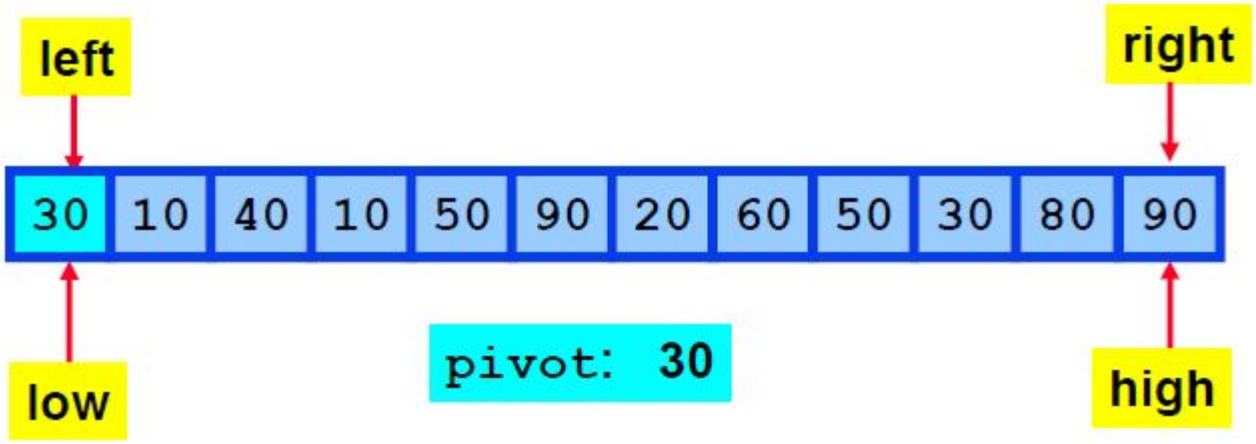
Quick Sort


pivot
18
12
15
23
42
38
36
29
27
pivot: 23
Recursively sort left half
Recursively sort right half

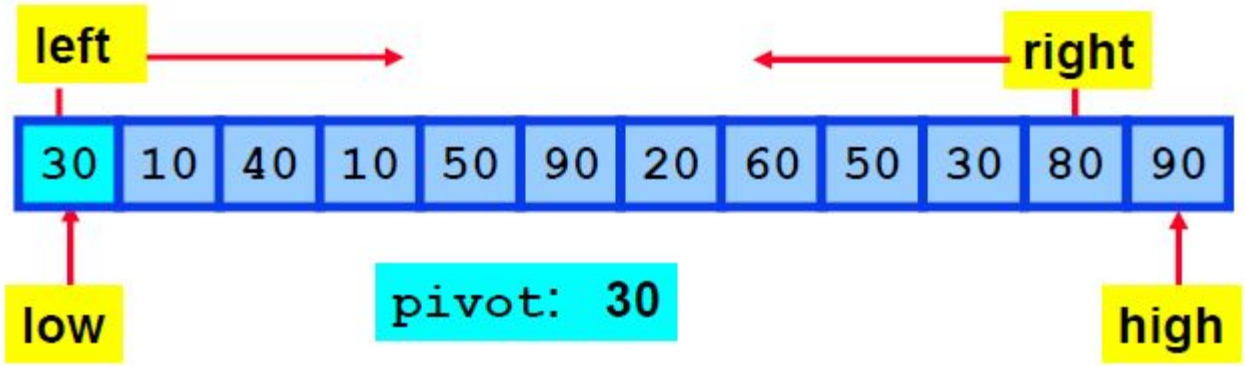
Another Illustration



Set left and right markers

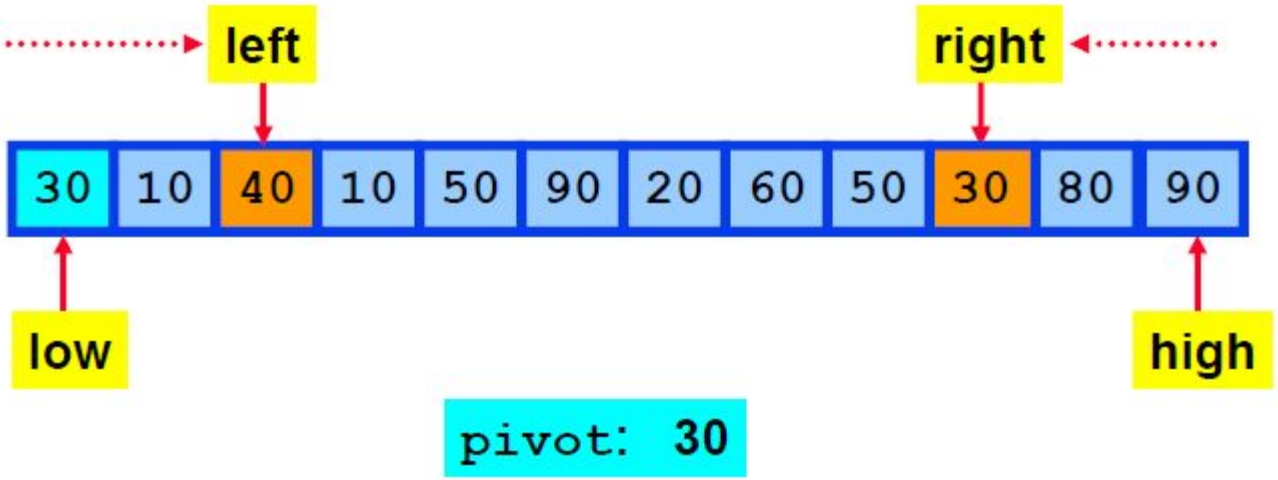


Move the markers
until they cross over

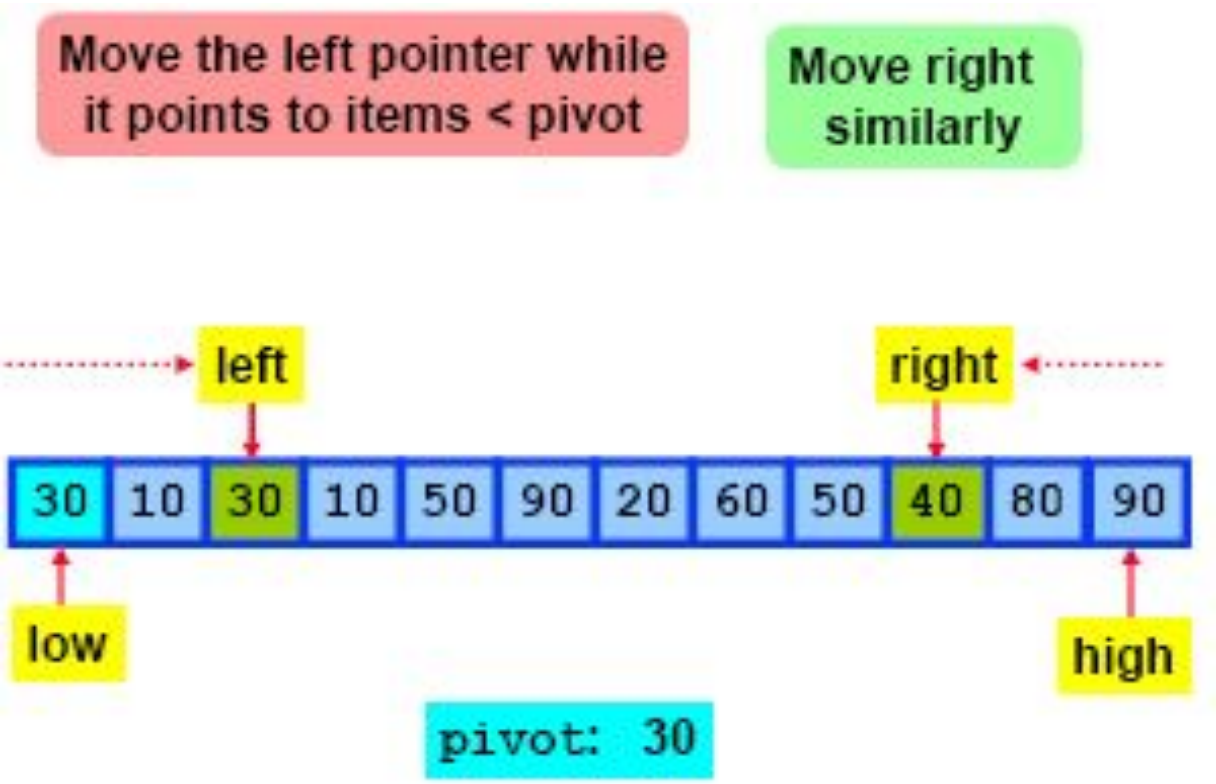


Move the left pointer while it points to items <= pivot

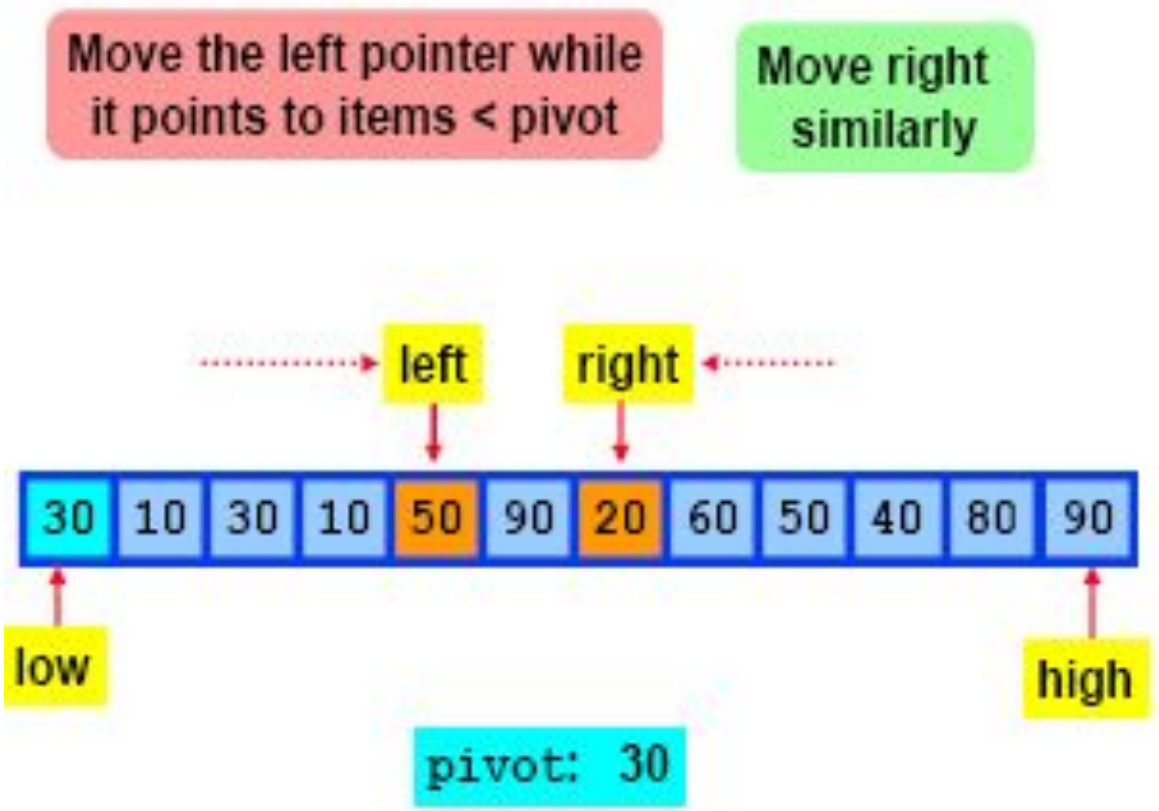
Move right similarly

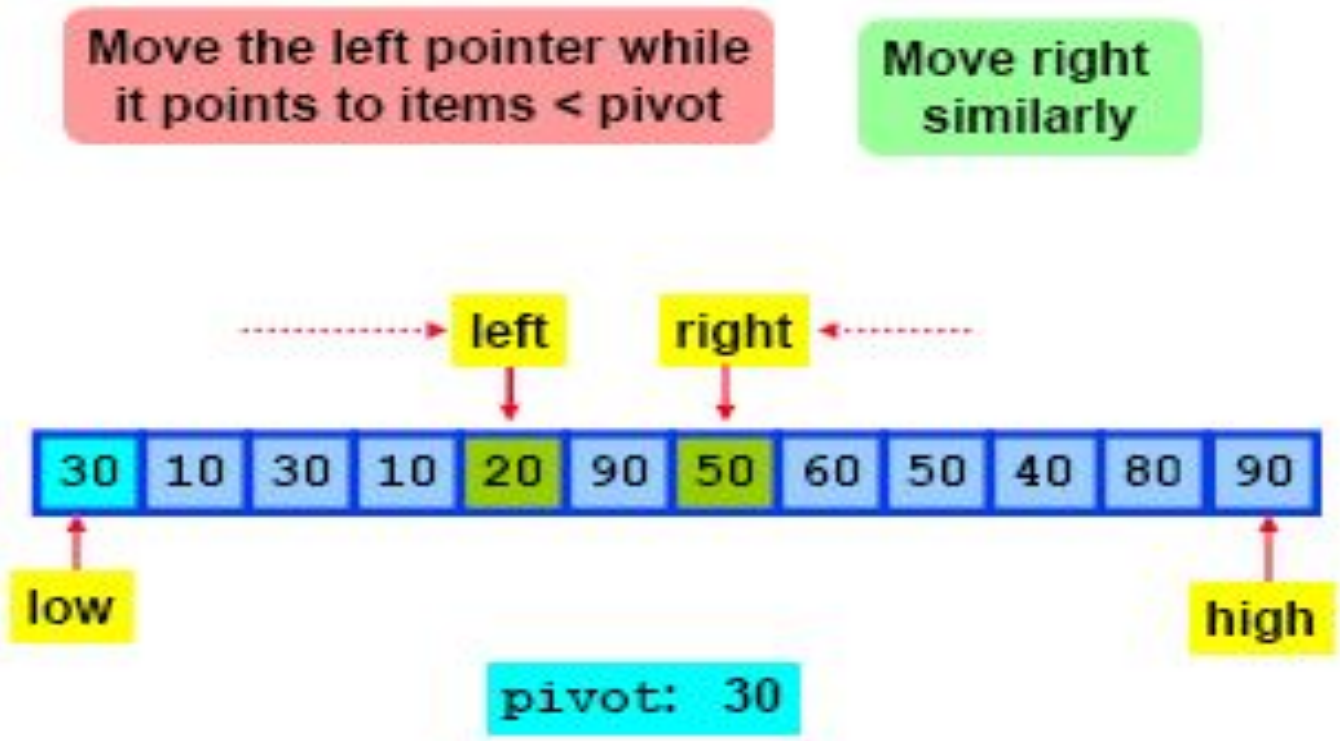


Another Illustration



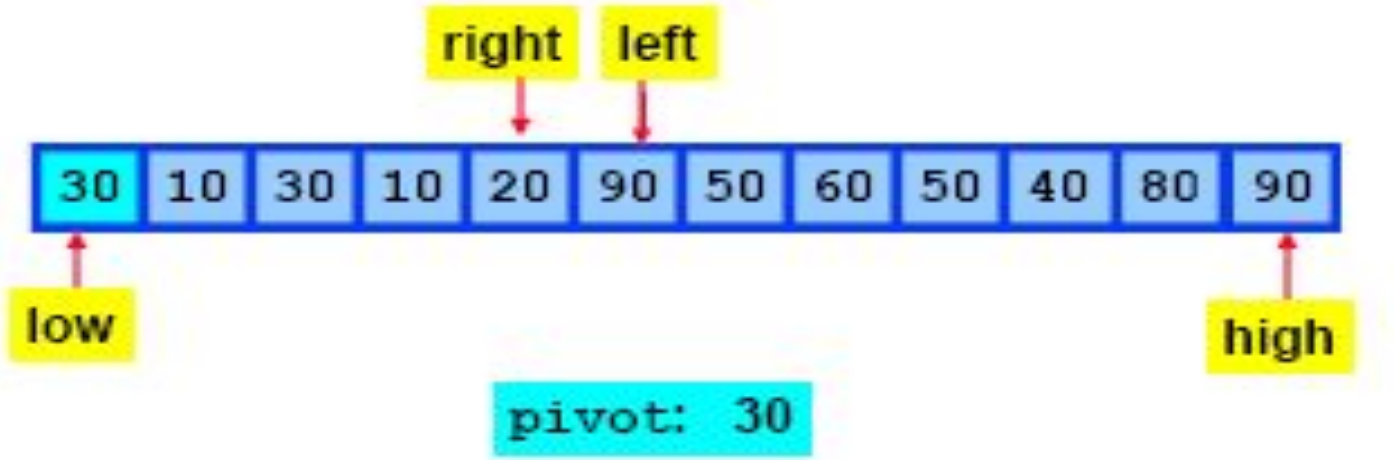
Another Illustration

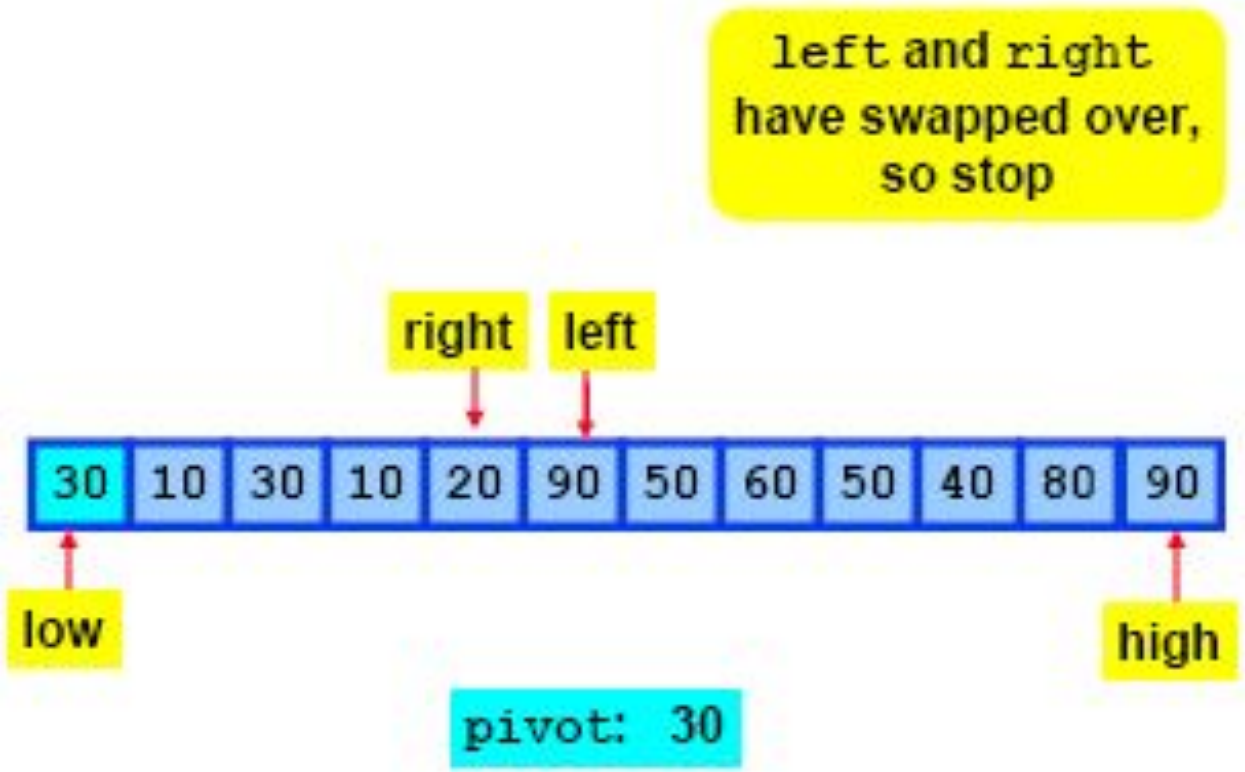


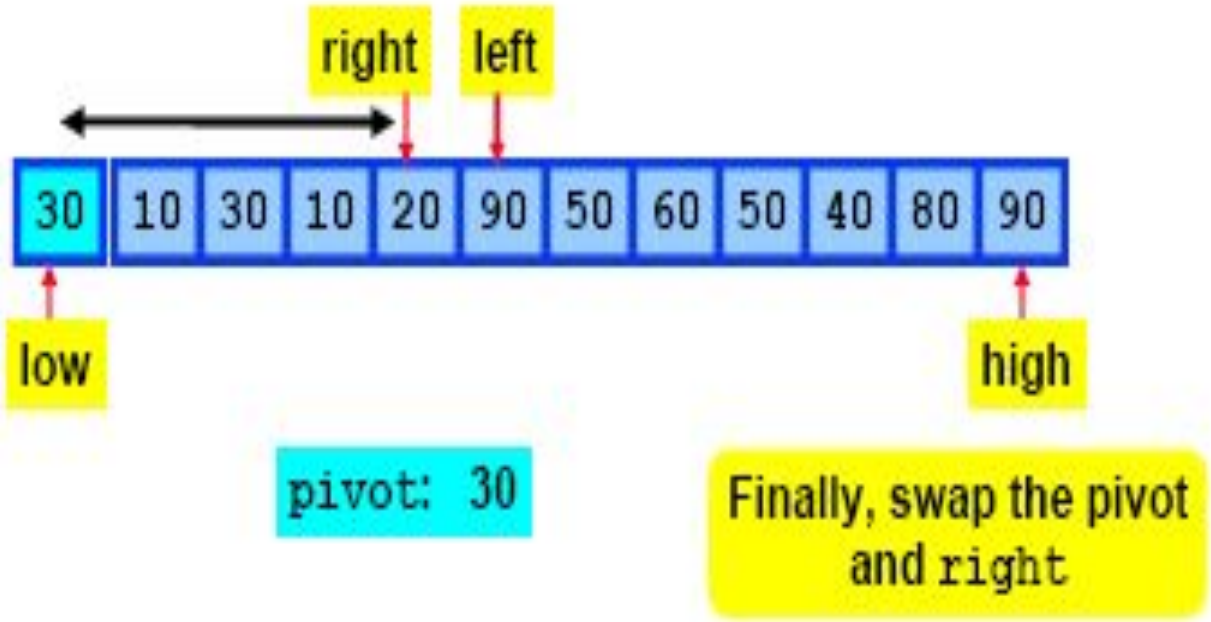

Move the left pointer while it points to items < pivot
Move right similarly
left
right
30 10 30 10 20 90 50 60 50 40 80 90
low
high
pivot: 30

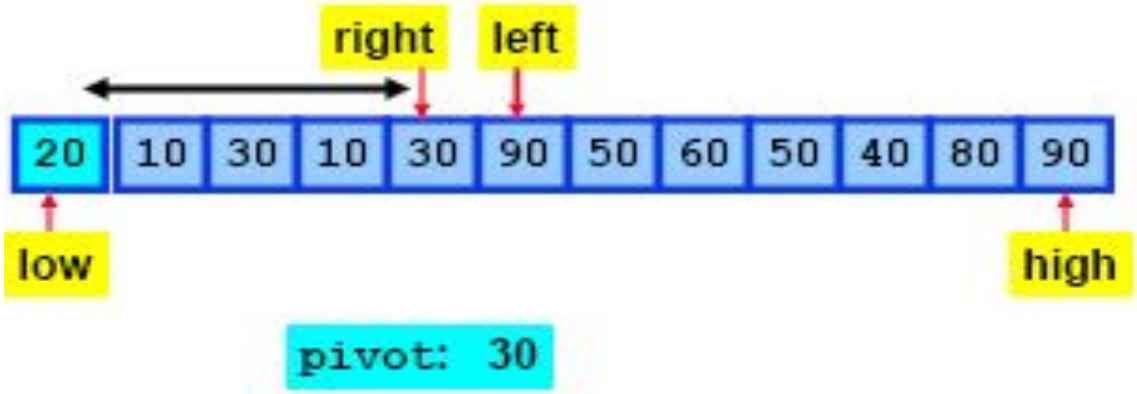
Move the left pointer while it points to items < pivot

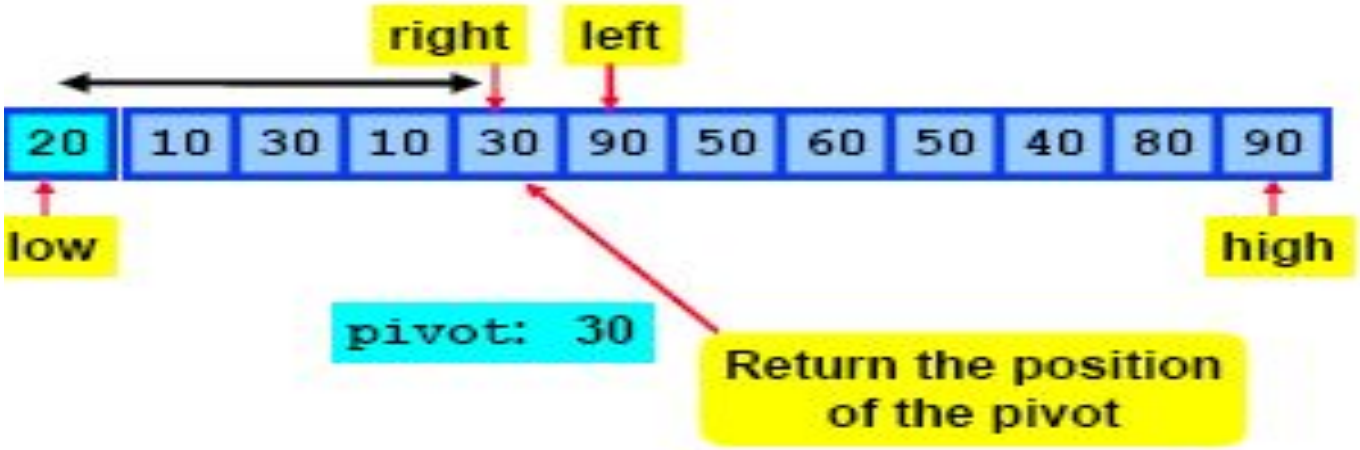
Move right similarly

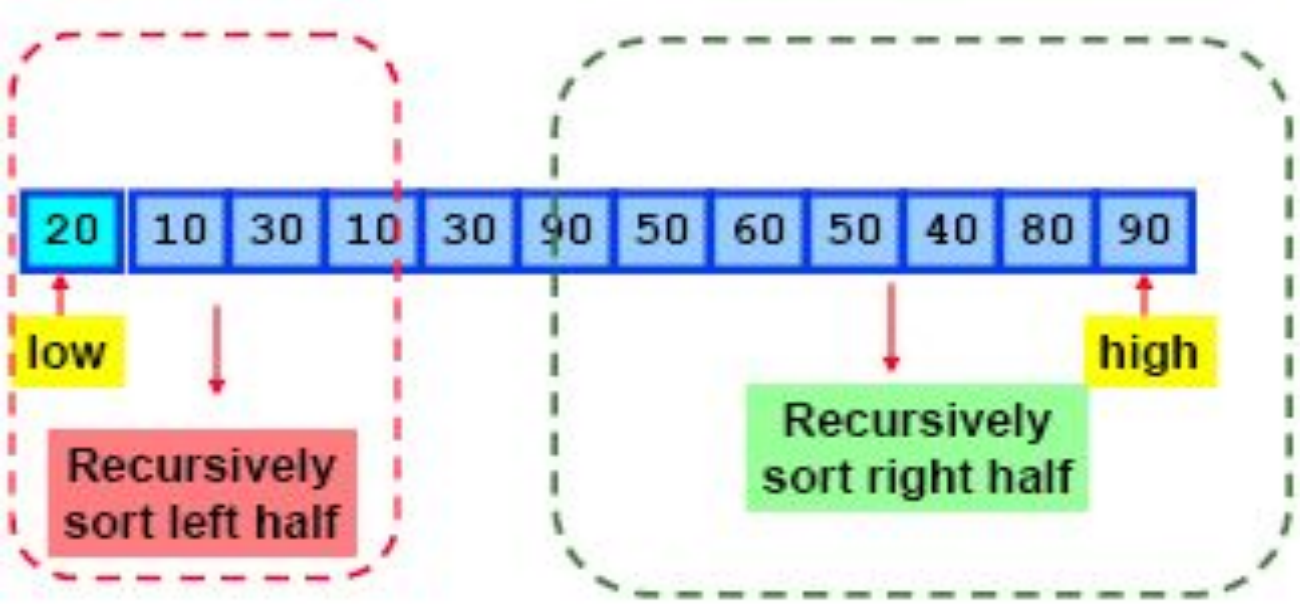



left and right have swapped over, so stop
right
left
30 10 30 10 20 90 50 60 50 40 80 90
low
high
pivot: 30


right
left
30
10
30
10
20
90
50
60
50
40
80
90
low
high
pivot: 30
Finally, swap the pivot and right


right
left
20
10
30
10
30
90
50
60
50
40
80
90
low
high
pivot: 30


right
left
20
10
30
10
30
90
50
60
50
40
80
90
low
high
pivot: 30
Return the position of the pivot


20
10
30
10
30
90
50
60
50
40
80
90
low
high
Recursively sort left half
Recursively sort right half