

# **SELECTION**



The problem is to **find the kth smallest element**.

Complexity worst case is  $T(n) = O(n^2)$  by repeated application of partition and the complexity is average case  $T(n) = O(n)$ .

Using median of medians as pivot complexity is  $T(n) = O(n^2)$ .

**COMPLEXITY  
WORST CASE  
PIVOT IS ARBITRARY  
 $T(n) = O(n^2)$**

**COMPLEXITY  
AVERAGE CASE  
 $T(n) = O(n)$**

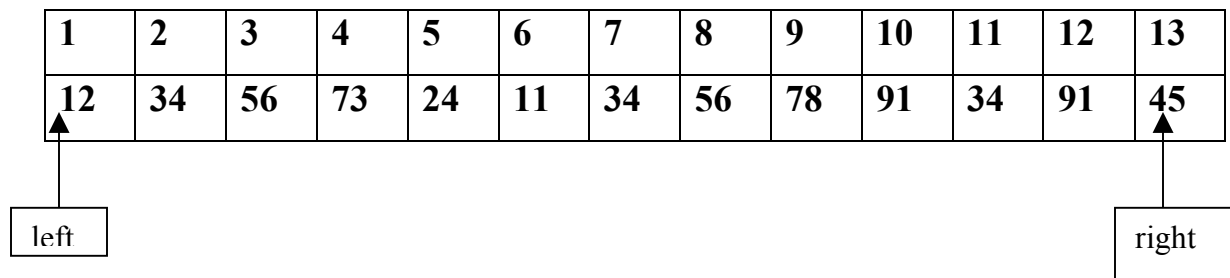
**COMPLEXITY  
WORST CASE  
PIVOT IS MEDIAN OF  
MEDIAN  
 $T(n) = O(n)$**

# PROBLEM FOR SIMULATION

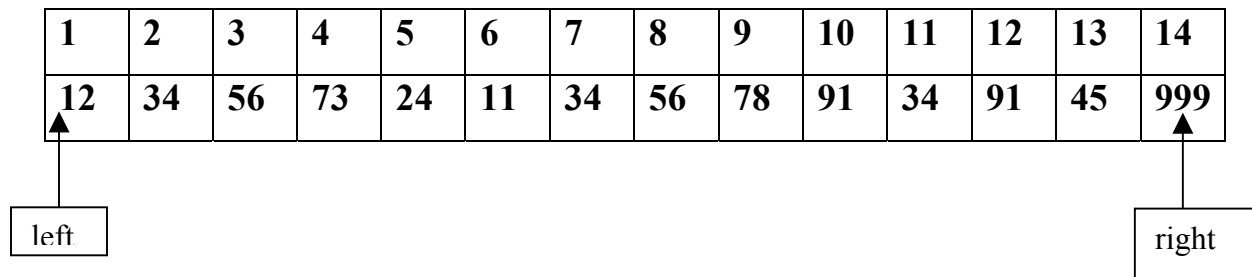
Consider the set {12,34,56,73,24,11,34,56,78,91,34,91,45}

FIND THE 5<sup>TH</sup> SMALLEST ELEMENT

INITIAL



KEEP A VERY LARGE ELEMENT AT THE RIGHT END



CALL PARTITION

The pointer “left” moves to the right skipping over smaller elements and the pointer “right” moves to the left skipping over larger elements.

**MOVE**

1	2	3	4	5	6	7	8	9	10	11	12	13	14
12	34	56	73	24	11	34	56	78	91	34	91	45	999

Diagram illustrating the initial state of an array during a partitioning step. The array contains 14 elements: 12, 34, 56, 73, 24, 11, 34, 56, 78, 91, 34, 91, 45, 999. The pivot element is 11, located at index 6. The pointer 'left' is positioned at index 2 (value 34), and the pointer 'right' is positioned at index 6 (value 11).

**SWAP**

1	2	3	4	5	6	7	8	9	10	11	12	13	14
12	11	56	73	24	34	34	56	78	91	34	91	45	999

Diagram illustrating the state of the array after a swap operation. The array contains 14 elements: 12, 11, 56, 73, 24, 34, 34, 56, 78, 91, 34, 91, 45, 999. The pointer 'left' is positioned at index 2 (value 11), and the pointer 'right' is positioned at index 6 (value 34).

**MOVE**

1	2	3	4	5	6	7	8	9	10	11	12	13	14
12	11	56	73	24	34	34	56	78	91	34	91	45	999

right

left

**POSITION FOR PIVOT ELEMENT 12 IS 2 i.e. 2 IS THE 2<sup>ND</sup> SMALLEST ELEMENT**

1	2	3	4	5	6	7	8	9	10	11	12	13	14
11	12	56	73	24	34	34	56	78	91	34	91	45	999

2<sup>nd</sup>

left

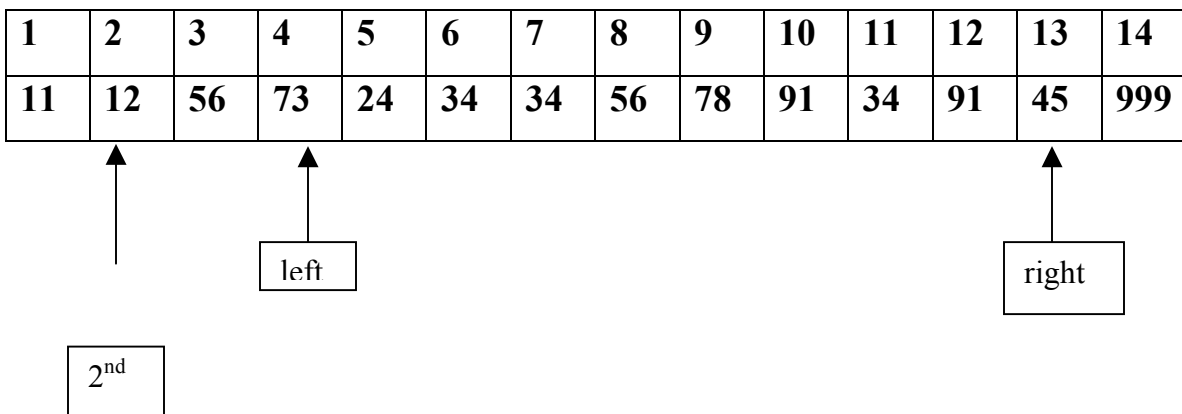
right

**SEARCHING TO THE RIGHT OF 2<sup>ND</sup> ELEMENT**

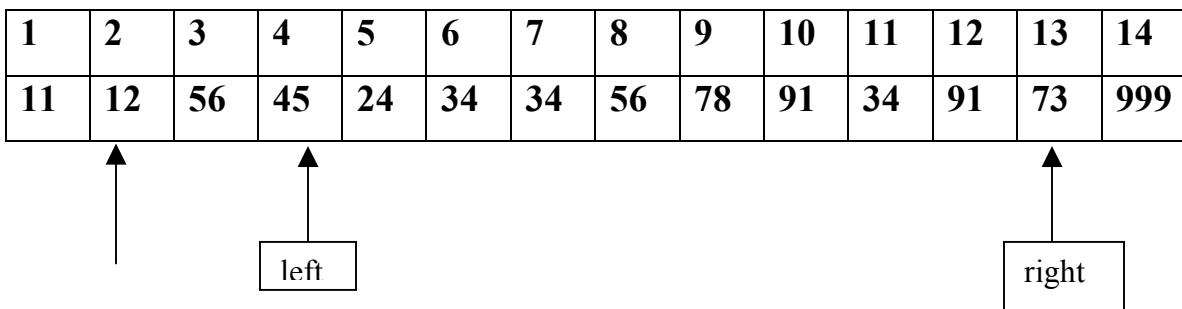
**CALL PARTITION**

The pointer “left” moves to the right skipping over smaller elements and the pointer “right” moves to the left skipping over larger elements.

### MOVE



### SWAP



2<sup>nd</sup>

### MOVE

1	2	3	4	5	6	7	8	9	10	11	12	13	14
11	12	56	45	24	34	34	56	78	91	34	91	73	999

↑

left

right

2<sup>nd</sup>

### SWAP

1	2	3	4	5	6	7	8	9	10	11	12	13	14
11	12	56	45	24	34	34	34	78	91	56	91	73	999

↑

left

right

2<sup>nd</sup>

## MOVE

1	2	3	4	5	6	7	8	9	10	11	12	13	14
11	12	56	45	24	34	34	34	78	91	56	91	73	999

Diagram illustrating the movement of pointers during a partitioning step. A box labeled "2<sup>nd</sup>" has an arrow pointing to the element 12 at index 2. A box labeled "right" has an arrow pointing to the element 34 at index 8. A box labeled "left" has an arrow pointing to the element 78 at index 9.

**POSITION FOR PIVOT ELEMENT HAS BEEN FOUND AS THE POINTERS HAVE CROSSED. THE 8<sup>TH</sup> SMALLEST ELEMENT IS 56.**

## SWAP RIGHT WITH PIVOT

1	2	3	4	5	6	7	8	9	10	11	12	13	14
11	12	34	45	24	34	34	56	78	91	56	91	73	999

Diagram illustrating the swap of the pivot element (78) with the element at the right pointer position (56). A box labeled "2<sup>nd</sup>" has an arrow pointing to the element 12 at index 2. A box labeled "right" has an arrow pointing to the element 56 at index 8. A box labeled "left" has an arrow pointing to the element 78 at index 9.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
11	12	34	45	24	34	34	56	78	91	56	91	73	999

↑
↑

2<sup>nd</sup>

8th

**SEARCH BETWEEN 3 AND 7**

1	2	3	4	5	6	7	8	9	10	11	12	13	14
11	12	34	45	24	34	34	56	78	91	56	91	73	999

↑
↑
↑
↑

2<sup>nd</sup>

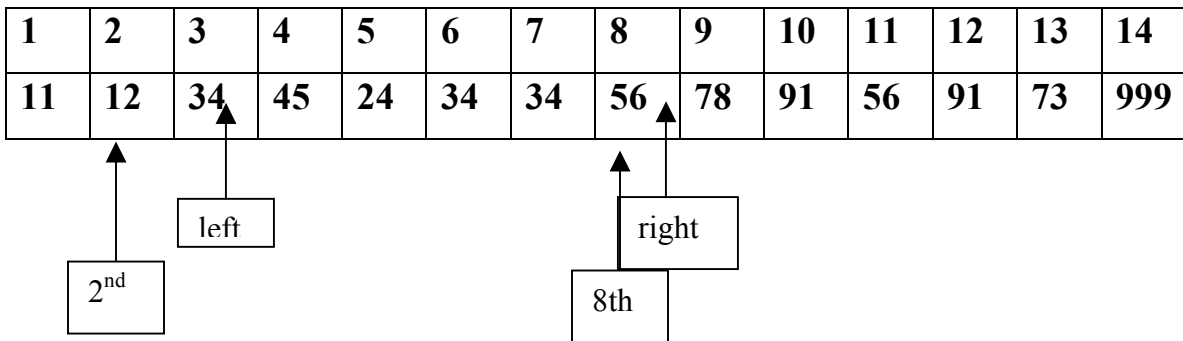
left

right

8th



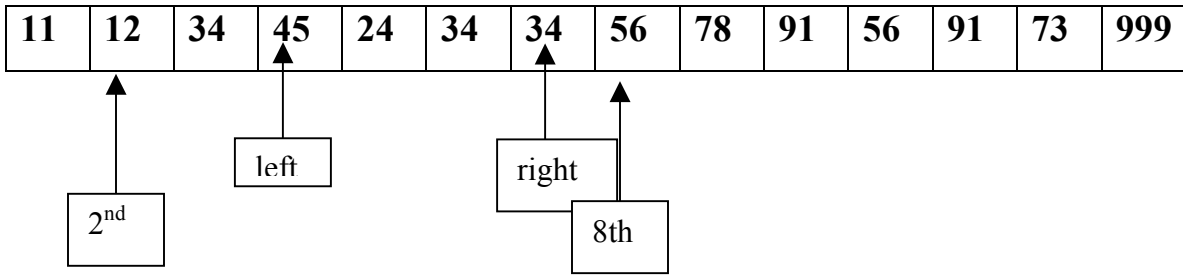
### CALL PARTITION (3,8)



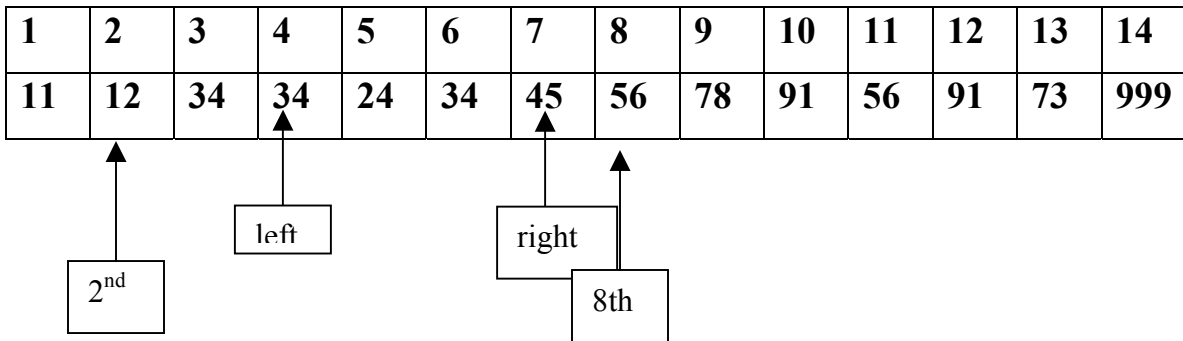
### MOVE

**The pointer "left" moves to the right skipping over smaller elements and the pointer "right" moves to the left skipping over larger elements.**

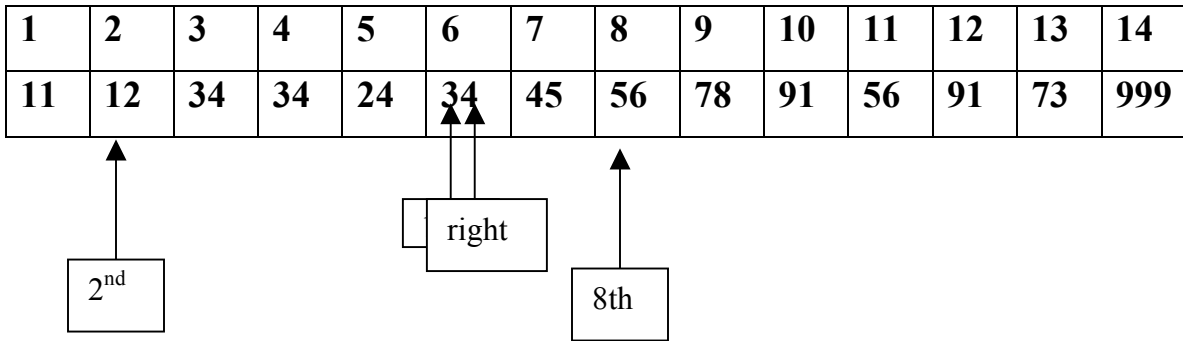
1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	----	----	----	----	----



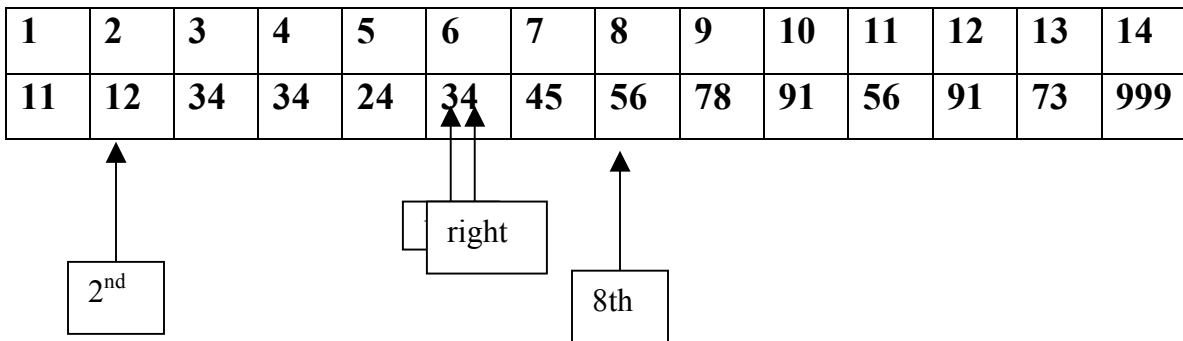
### SWAP



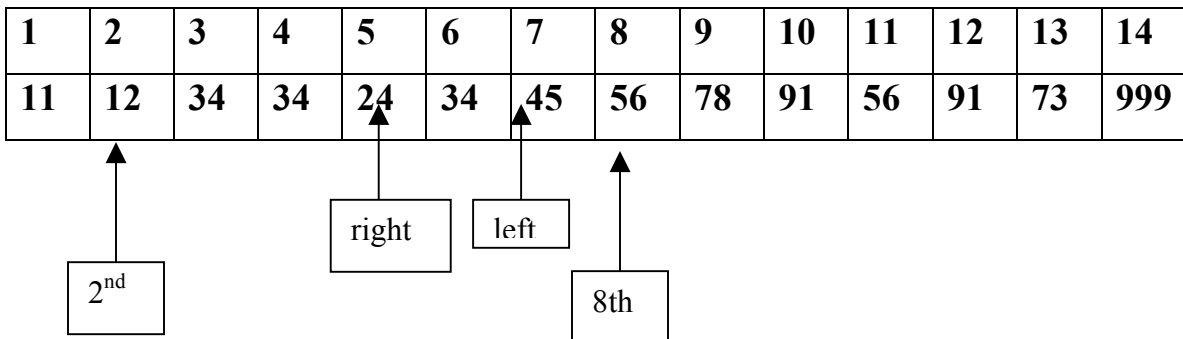
### MOVE



**SWAP**



**MOVE**



**POINTERS HAVE CROSSED SO 5 IS THE POSITION OF THE PIVOT ELEMENT 34.**

1	2	3	4	5	6	7	8	9	10	11	12	13	14
11	12	24	34	34	34	45	56	78	91	56	91	73	999

The diagram illustrates the state of pointers during a partitioning process. Three boxes, each containing a pointer position, have arrows pointing to specific elements in the array. The first box, labeled '2<sup>nd</sup>', points to the element '12' at index 2. The second box, also labeled '2<sup>nd</sup>', points to the element '34' at index 5. The third box, labeled '8<sup>th</sup>', points to the element '56' at index 8.

**POINTERS HAVE CROSSED SO 5 IS THE POSITION OF THE PIVOT ELEMENT 34.**