

Merge sort

1. Divide array in two parts
2. Sort both partitions individually (by merge sort only)
3. Merge sorted partitions into temporary array
4. Overwrite temporary array into original array

$$\text{find mid} = \frac{l+r}{2}$$

$$\text{sizeof(temp[])} = r - l + 1$$

$$8 - 0 + 1 = 9$$

$$4 - 0 + 1 = 5$$

$$8 - 5 + 1 = 4$$

left partition : $l \rightarrow m$

right partition : $m+1 \rightarrow r$

Number of elements = n

levels of division = $\log n$

Comparisons per level $\approx n$

Total comps = $n \log n$

Time \propto comps

Time $\propto n \log n$

Best
Avg
worst

$$T(n) = O(n \log n)$$

temp[] - needed to merge two sorted partitions

Auxilliary Space $\propto n$

$$S(n) = O(n)$$

6	1	9	7	3	8	2	4	5
0	1	2	3	4	5	6	7	8

l m r

6	1	9	7	3
0	1	2	3	4

8	2	4	5
5	6	7	8

1	3	6	7	9
0	1	2	3	4

2	4	5	8
5	6	7	8

temp	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8

Merge sort

1	2	3	4	5	6	7	8	9
+	8	6	7	9	2	4	5	8
1	6	9						
x	6		3	7	2	8	4	5
6	1	8	7	3	8	2	4	5
0	1	2	3	4	5	6	7	8

$ms(arr, 0, 8) \quad m=4$

1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8

$ms(arr, 0, 4) \quad m=2$

1	3	6	7	9
0	1	2	3	4

$ms(arr, 0, 2) \quad m=1$

1	6	9
0	1	2

$ms(arr, 0, 1) \quad m=0$

1	6
0	1

6
0

1
1

$ms(arr, 2, 2)$

9
2

$ms(arr, 3, 4) \quad m=3$

3	7
0	1

$ms(arr, 3, 3)$

7
3

$ms(arr, 4, 4)$

3
4

$ms(arr, 5, 8) \quad m=6$

2	4	5	8
0	1	2	3

$ms(arr, 5, 6) \quad m=5$

2	8
0	1

$ms(arr, 5, 5)$

8
5

$ms(arr, 6, 6)$

2
6

4	5
0	1



- i. extreme left or right element
- ii. middle element
- iii. median
 - random 3 elements
 - random 5 elements
- iv. dual pivot

Best
Avg

$$T(n) = O(n \log n)$$

11	22	33	44	55	66
	22	33	44	55	66
		33	44	55	66
			44	55	66
				55	66
					66

level $\approx n$

total comps = n^2

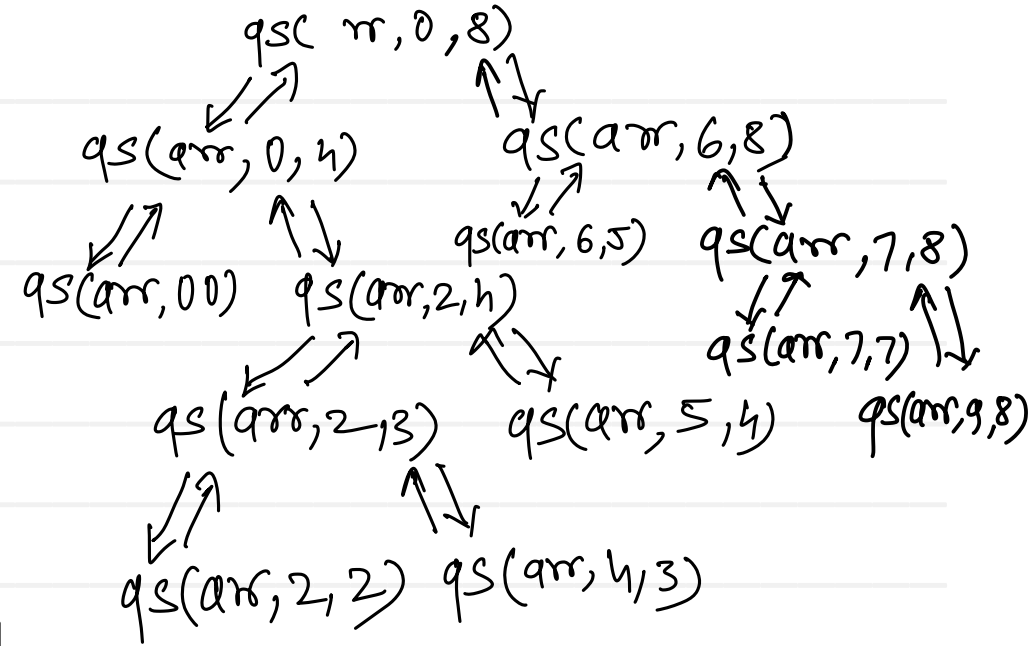
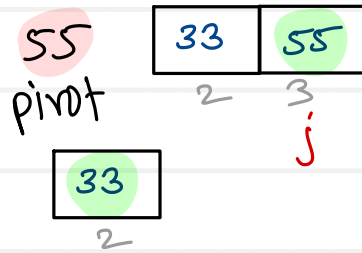
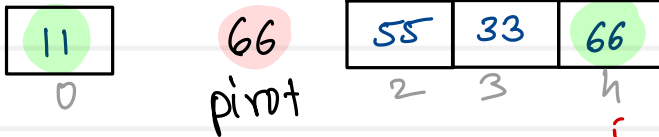
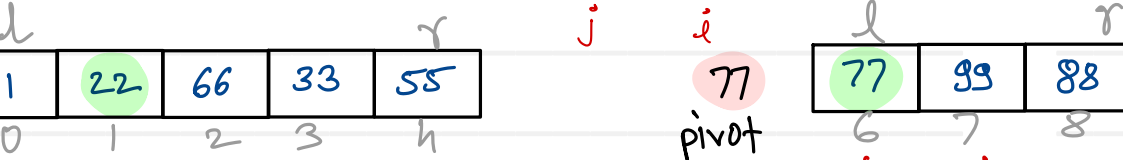
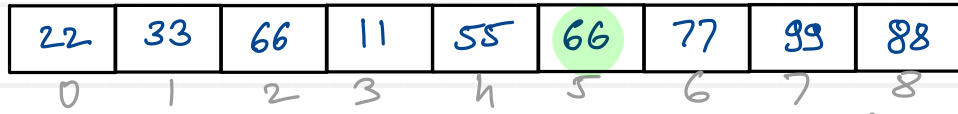
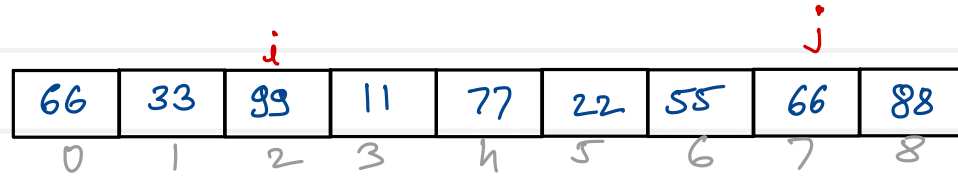
Time $\propto n^2$

Worst $T(n) = O(n^2)$

- time complexity of quick is dependent on selection of pivot.



Quick sort



	space	Best	Time Avg	Worst
selection sort	$O(1)$ in place sorting algorithm	$O(n^2)$	$O(n^2)$	$O(n^2)$
bubble sort		$O(n)$	$O(n^2)$	$O(n^2)$
insertion sort		$O(n)$	$O(n^2)$	$O(n^2)$
Heap sort		$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Quick sort		$O(n \log n)$	$O(n \log n)$	$O(n^2)$
Merge sort	$O(n)$	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$

33_a 11_a 22 11_b 33_b

11_a 11_b 22 33_a 33_b — more stable

11_b 11_a 22 33_b 33_a — unstable



Thank you!!!

Devendra Dhande

devendra.dhande@sunbeaminfo.com