1 5 10 — 10 11 17

2 6 11 — 15 —

15 16 17

m

n

1 100

2 200

5 300

TC → O(n log m).

3 30 (0, m-1) 33 60 69 51 51 36 40

Loop : 36

3 30 → 33 → 60 → 69 51 36 40

Loop = 39

TC = O(n + m)

AS = O(1)

[5a, 10, 3, 5b, 6, 2]

Stable sorting algo → [2, 3, 5a, 5b, 6, 10] (Always).

Unstable sorting algo → [2, 3, 5b, 5a, 6, 10] OR [2, 3, 5a, 5b, 6, 10]

Merge Sort

0 — 25 — 50

[ ]

0 — 12 — 25 — 37 — 50

[ ] [ ]

0 — 6 — 12 — 18 — 25 — 31 — 37 — 43 — 50

[ ] [ ] [ ] [ ]

n m

[1, 3, 10, 12] [2, 3, 5, 60]

[1, 2, 3, 3, 5, 10, 12, 60]

Merging → O(n+m).

For merge sort → $O\left(\frac{n}{2} + \frac{n}{2}\right) = O(n)$.

low — high

arr: [|]

temp: [ ]

low — high

arr: [1, 3, 10, 12, 2, 3, 5, 60]

temp: [1, 2, 3, 3, 5, 10, 12, 60]

[1, 3, 10, 12] [2, 3, 5, 60]

[1, 2, 3, 3, 5, 10, 12

[0 1] mid: 0

T(n) = 2T(n/2) + n

M = log(n)

T(n) — n

T(n/2) T(n/2) — n

T(n/4) T(n/4) T(n/4) T(n/4) — n

TC = O(n log n).

AS = O(n)

A: [1, 2, 5, 6] — a

B: [1, 3, 6] — b

C: [8, 11] — c

O(a+b). to merge A[] & B[].

TC O(a+b+c) Final merge.

AS = O(a+b)

A: [1, 2, 5, 6] –

B: [1, 3, 6] –

C: [8, 11] –

[1, 1, 2, 3, 5, 6, 6, 8, 11]

TC = O(a+b+c).

AS = O(1).

(i & j & k) (i || j || k).

(i j)

(i k) ⇒

(j k)

(i)

(j)

(k)

k/2

k/2.

---

arr: 900, 910, 920, 1100, 1500, 1800

dof: 910 → 1, 1200 → 1, 1120 → 2, 1130 → 3, 1900 → 1, 2000 → 2

- 900
- 950

OA = 3

DF → TC = O(n²)

AS → O(n)

950 - 1100

900 - 910

1

950 - 1130

2

---

a d a a a d d d a a a d

900 910 920 950 1100 1120 1130 1150 1800 1900 1900 1900

1 0 1 2 3 2 1 0 2.1 3 2 1 0

TC → O(n log n)

AS → O(n).

arr: 900, 910, 920

dof: 910, 920.

OA = 2.

a a d d

900 910 910 920.

1 2 1 0