

Divide-and-Conquer

Monday, February 8, 2021 12:11 PM

Merge Sort

5	3	2	4	6	9	11	13	0,7
---	---	---	---	---	---	----	----	-----

5	3	2	4	6	9	11	13	0,3
---	---	---	---	---	---	----	----	-----

5	3	2	4	6	9	11	13	0,1
---	---	---	---	---	---	----	----	-----

5	3	2	4	6	9	11	13
---	---	---	---	---	---	----	----

Ms(0,1) calls ms(0,0)

Returns to ms(1,1)

Left -> Right -> Root (postorder)

Merge(0,0,1) //sorts

out-place sorting because auxiliary array created in merge() to sort

3	5	2	4	6	9	11	13
---	---	---	---	---	---	----	----

Ms(2,3) , merge(2,2,3)

3	5	2	4	6	9	11	13
---	---	---	---	---	---	----	----

Merge(0,1,3)

2	3	4	5	6	9	11	13
---	---	---	---	---	---	----	----

$$T(n) = 2 * T\left(\frac{N}{2}\right) + c * n, T(1) = 1$$

Try minimizing copying from a to b

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

By storing the numbers in a linked list instead of an array, n may reduce to 1



If you work on one half at a time, best case the storage $O(n) \rightarrow O(\log n)$

Min Max Problem

```
a(0) = min = max
for(int i=1; i<n; i++)
    If(a[i]<min) min=a[i]
    If(a[i]>max) max=a[i]
```

Replacing with else does not change complexity --> brute force algorithm --> $O(n)$

Recursive:

14	15	13	2
----	----	----	---

14	15
Min	Max

13	2
Max	Min

Merge:

2<14	15>13
Min	Max

2 steps only

But we're working on both halves --> $1.5n - 2$

$$T(1) = 0$$

$$T(2) = 1$$

$$T(n) = 2 * T\left(\frac{n}{2}\right) + 2$$