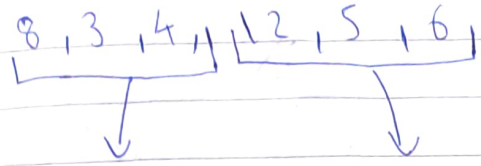
* Merge sort



3, 4, 8 5, 6, 12

After sorted merge

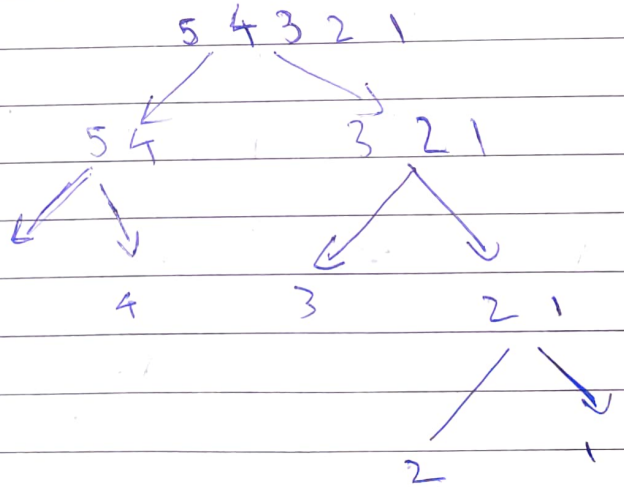
[3, 4, 5, 6, 8, 12]

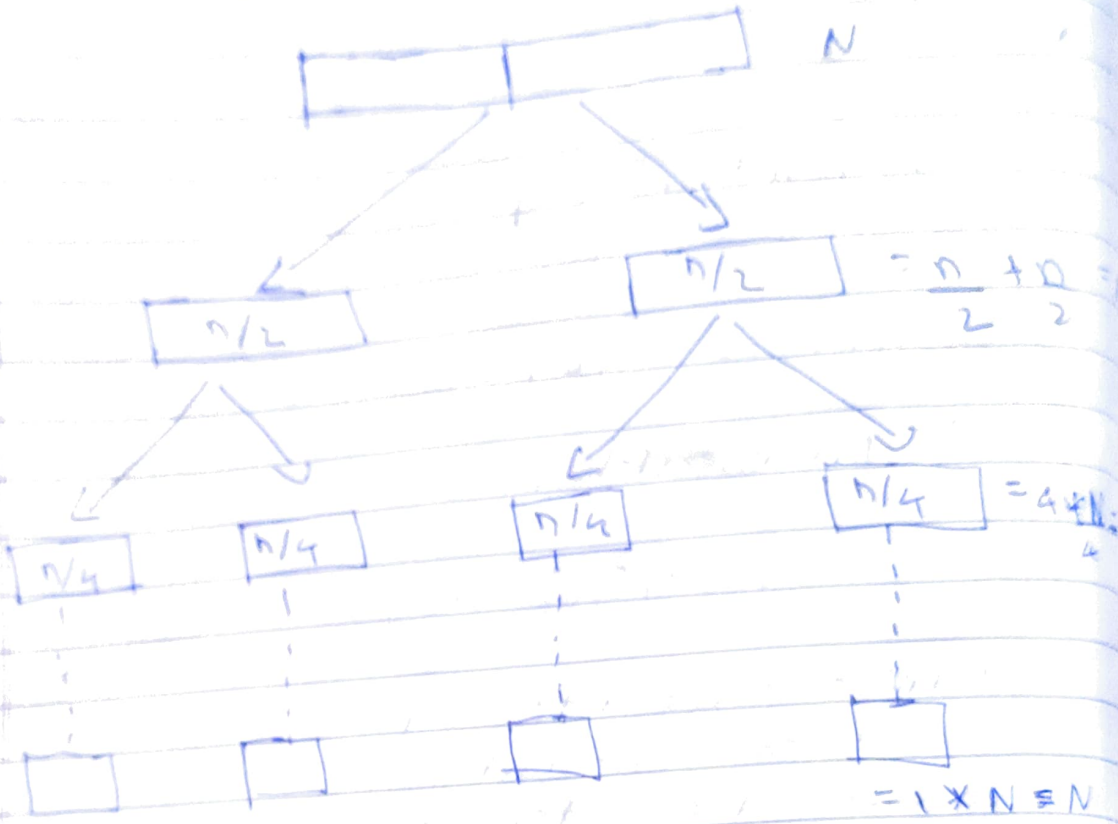
① Divide array into 2 parts
② Get both part sorted via recursion
③ merge the sorted parts

Explain ③ : arr1 = [3, 5, 9, 19, 32]

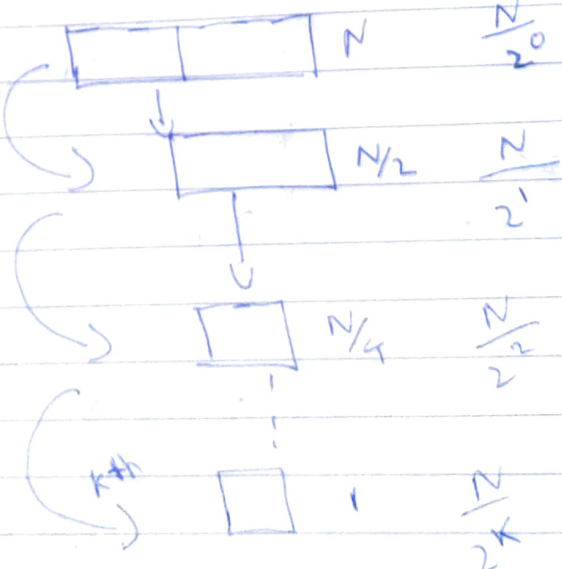
arr2 = [4, 6, 8]

size (arr1 + arr2) [3, 4, 5, 6, 8, 9, 19, 32]





* At every level, N elements are being merged



$1 = \frac{N}{2^K} \Rightarrow 2^K = N$

$K \log 2 = \log N$

$$K = \log_2 N$$

$$\therefore O(N * \log_2 N)$$

Space complex.
Aux = $O(N)$

- By akrabazzi formulae

Recurrence relation

$$O(T) = O(T/2) + O(T/2) + (N-1)$$

$$= 2O\left(\frac{T}{2}\right) + N - 1$$

$$2 \times \frac{1}{2}^{p} = 1$$

$$p = 1$$

$$O\left(x^p + x^p \int_1^x \frac{g(u)}{u^{p+1}}\, du\right)$$

$$u^{-2} \to u^{-1}$$

$$= O\left(x + x \int_1^x \frac{u-1}{u^2}\, du\right)$$

$$= O\left(x + x\left[\frac{1}{u} - \frac{1}{u^2}\right]_1^x\right)$$

$$= O\left(x + x\left[\log u + \frac{1}{u}\right]_1^x\right)$$

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

5, 4, 3, 2, 1

Pivot: Choose any element → after 1st pass all the elements < p will be on L.Hs & elements > p will be at R.Hs of p

5, (4), 3, 2, 1

1, 3, 2, 4, 5

1, 3, 2 | 5

1, 2, 3

★ how to put pivot at correct position?

s p e

5, (4), 3, 2, 1 p = 4

s e

1, 4, 3, 2, 5

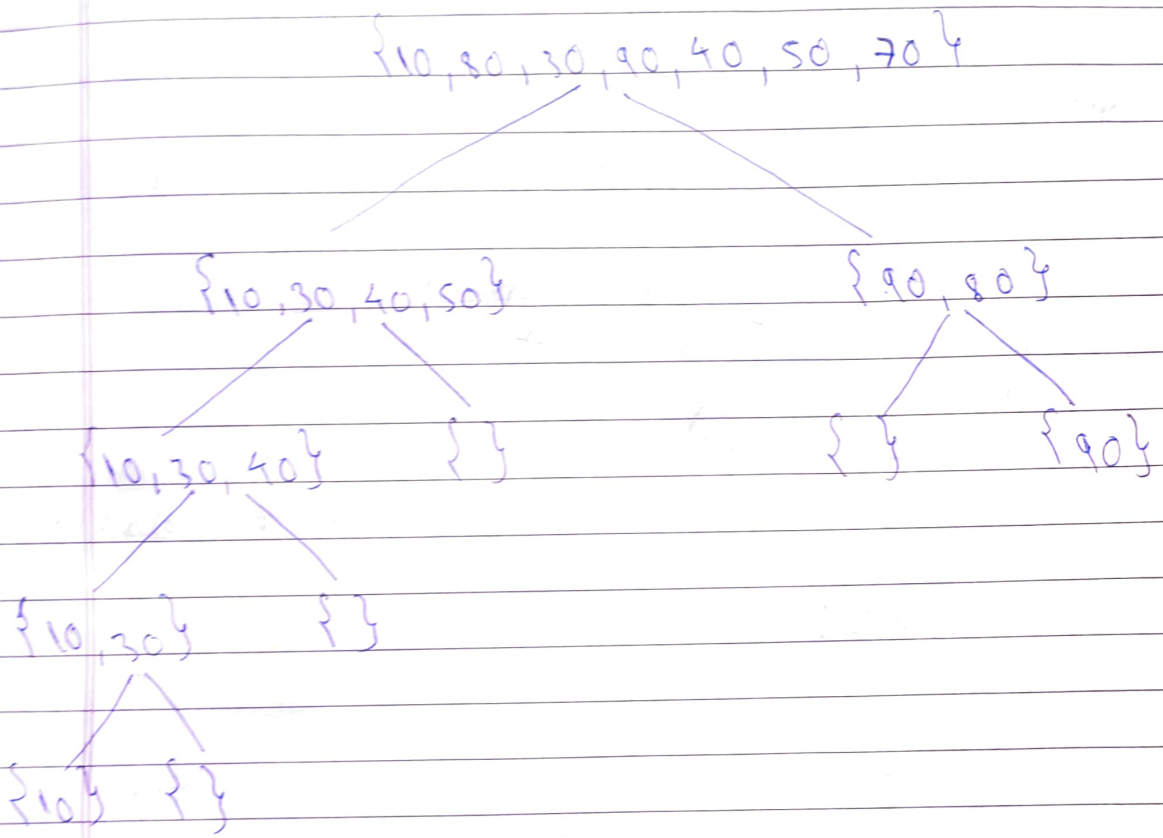
l e s h

1, 2, 3, (4), 5 → (s, high)

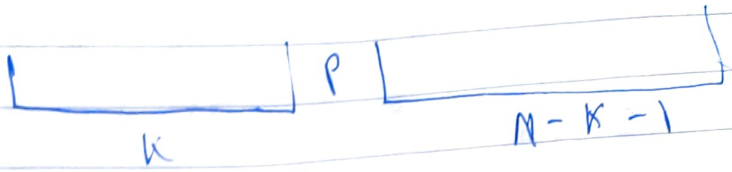
(low, end)

while n[s] < p;
s++
while n[e] > p;
e--

four variable → swapping
s & end
low & high → main tells us which part of array you are working on.



* How to pick pivot:
  * Random element
  * corner element
  * Pick the middle element



Important

$$T(N) = T(K) + T(N-K-1) + O(N)$$

Worst case:

$K = 0$ 3, 5, 8, 9, 20, 34, 18, 66

$$T(N) = T(N-1) + O(N)$$

$$= O(N^2)$$

linear recurrence relation

Best case:

$K = N/2$

$$T(N) = T\left(\frac{N}{2}\right) + T\left(\frac{N}{2}\right) + O(N)$$

$$T(N) = 2T\left(\frac{N}{2}\right) + O(N)$$

$O(N \log N)$

★ Notes:-

- Not stable (already covered)
- In-place ∴ that's why preferred for arrays instead of MS;
  MS takes O(N) extra space
- MS is better in linked lists due to memory allocation → not continuous

★ Hybrid sorting algorithms (Tim sort):

merge sort + Insertion sort

↓

works well with partially sorted data

l — p — h

~~5 2 3 4 5~~

l p h

5 4 3 2 1

s p