Hashmap

25 September 2022 13:07

store key value pair.

Roll no. <Int>	Name <String>
1	Rahul
2	Sarav
5	Manish
10	

unique

food item <String>,

	price <Int>
Maggi	40
Pizza	100
Burger	60
Coke	40

O(1)

Intersection of two arrays

25 September 2022 13:15

arr1[ ] = [1, 4, 6, 3, 4, 1, 7] (N)

arr2[ ] = [0, 1, 6, 3, 1, 6] (M)

(N×M)

(N×N)

output[ ] = [1, 1, 3, 6]

key int	value int
1	~~1~~ ~~1~~ 1
4	~~1~~ 2
6	~~1~~ 0
3	~~1~~ 0
7	1

[1, 6, 3, 1]

N+N = 2N = O(N) time

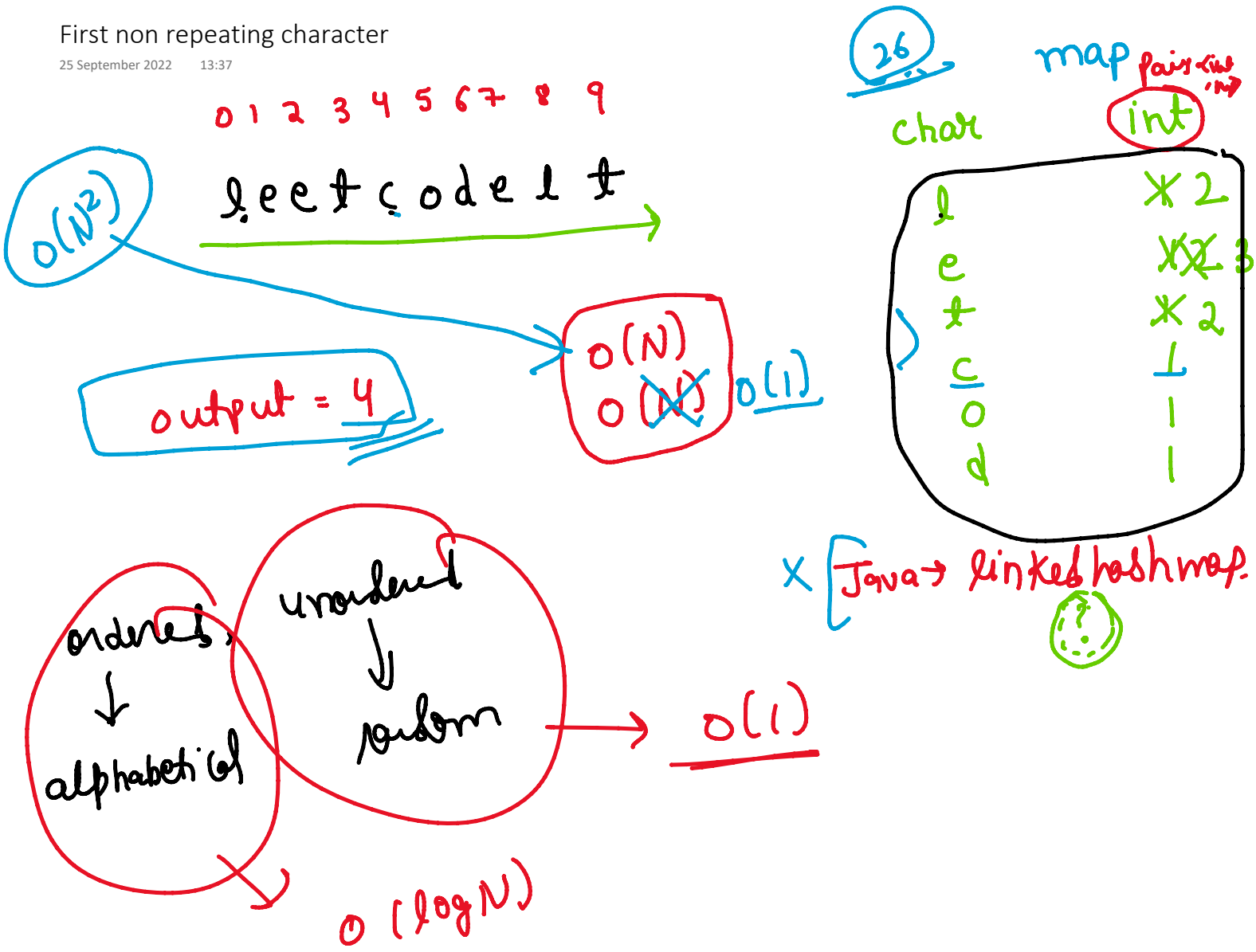
= O(N) space

[2, 3, 3, 4, 5, 5] sorted.

output [2, 4]

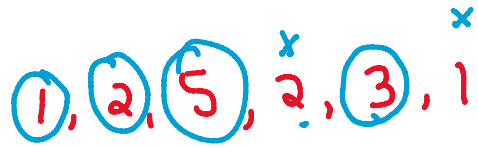
First non repeating character

25 September 2022 13:37



Subarray with sum 0

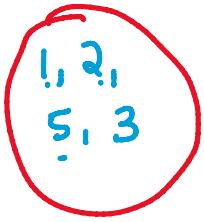
25 September 2022 13:55



set

unique elements

set <int>



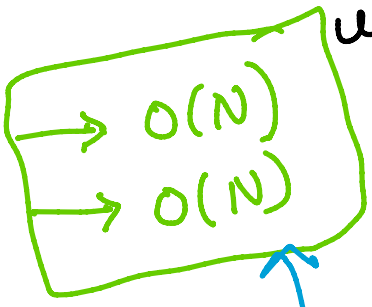
3 -2 5 1 4 -7 -3 8 — true

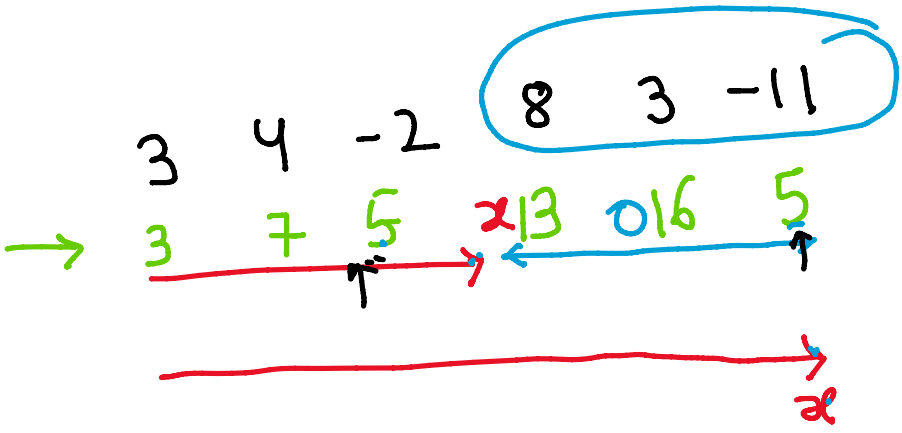
-3 -2 6 1 7 -1 — false

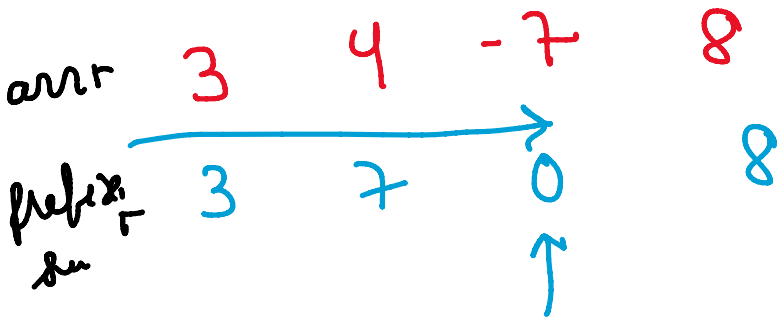
5 -4 8 3 2 -5 4 — true

Prefix Sum: 5 1 9 12 14 9

using set.

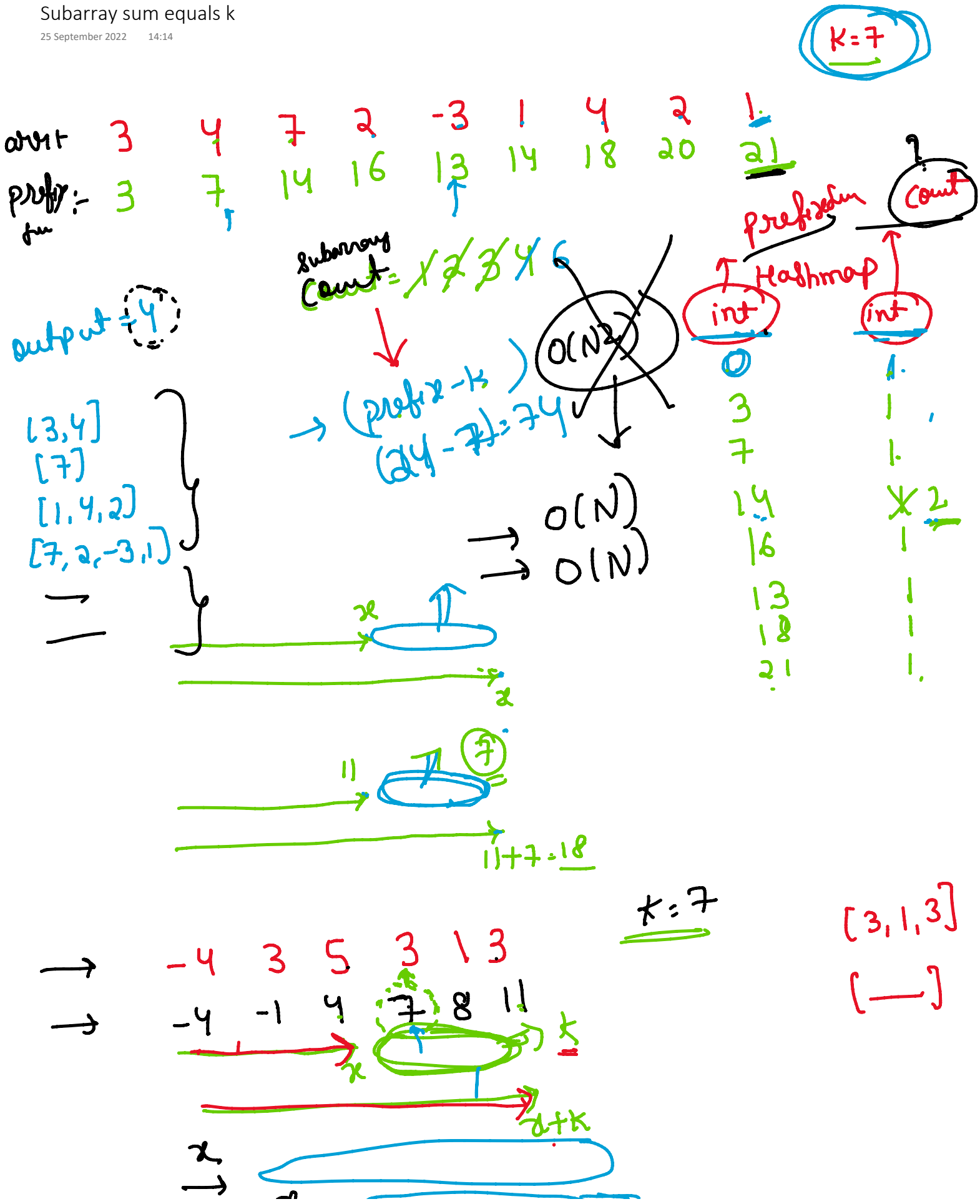





arr
3
4
-7
8
prefix sum
3
7
0
8

Subarray sum equals k

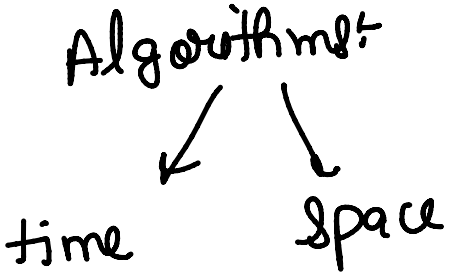
25 September 2022 14:14




x
x
x
a+k

Time and Space Complexity

25 September 2022 15:07



→ decrease time.

$O(N^2)$, $O(1)$

↓ ↓

$O(N)$ $O(N)$

$3n^2 + 2n \Rightarrow O(N^2)$

$\frac{n}{5} + \log N \Rightarrow O(N)$

$O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(n^3)$, ...

$O(2^n)$, $O(n!)$

$2n + 100 \log N \Rightarrow O(n)$

→ Bubble sort
→ Binary search
→ Recursive

→ Recursive

→ Merge sort

Bubble sort $O(N^2)$

5, 2, 4, 3, 0, 1

- - - . 5 — 1st iteration

- - - . 4 5 — 2nd iteration.

$N-1 + N-2 + N-3 + N-4 + \cdots\cdots + 1$

$1 + 2 + 3 + 4 + \cdots\cdots + (N-1)$

$= \frac{N(N-1)}{2} \times K$

$= \frac{N^2}{2}K - \frac{N}{2}K \Rightarrow O(N^2)$

Recurrence

$T(n) = K \cdot n + T(n-1)$

$T(n-1) = K(n-1) + T(n-2)$

$T(n-2) = K(n-2) + T(n-3)$

$\vdots$

$T(1) = K(1) + 0$

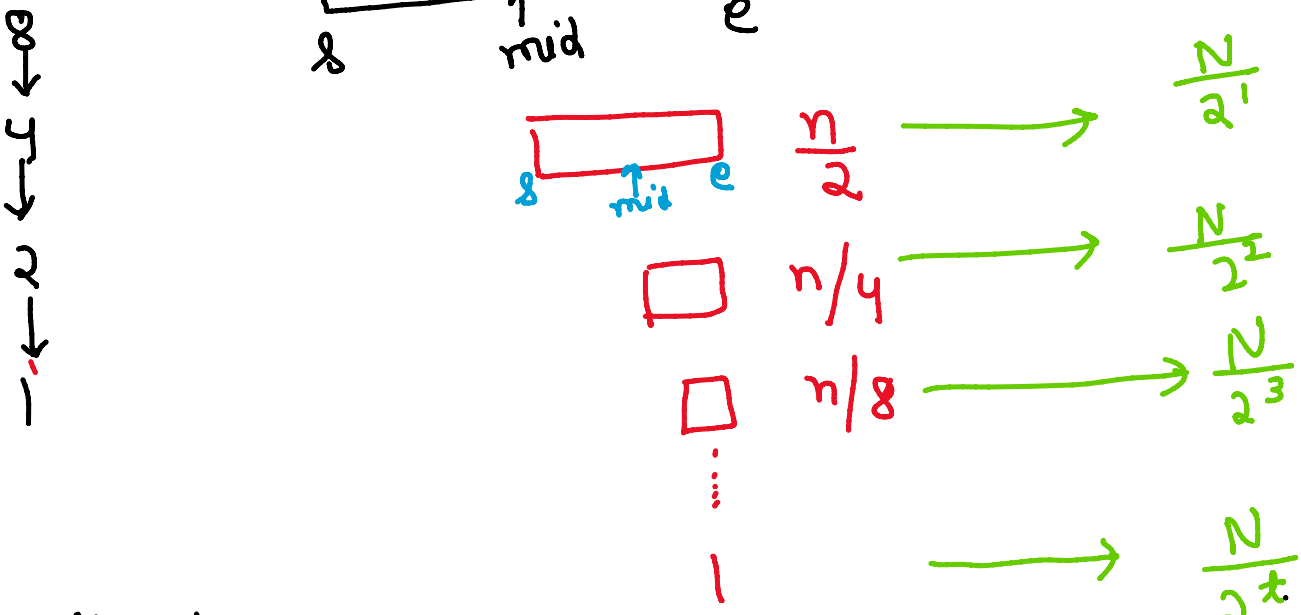
$T(n) = K(n + (n-1) + (n-2) + \cdots 1)$

$$T(n) = k(n + (n-1) + (n-2) + \dots 1)$$

$$= k\,\frac{(n)(n-1)}{2}$$

$$= O(N^2)$$

Binary search $O(\log N)$



$$\frac{N}{2^t} = 1$$

$$N = 2^t$$

$$\log_2 N = \log_2 2^t$$

$$\log_2 N = t$$

$O(\log N)$

Recurrence

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

n, $n/2$, $n/4$, ... ✓

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

$T(n/2) = k + T(n/4)$

$T(n/4) = k + T(n/8)$

$\vdots$

$1 = k + 0$

$n, n/4, n/8, n/16, \ldots, 1$

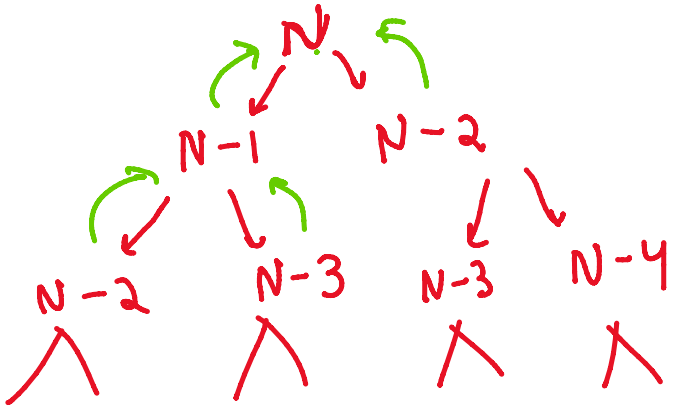
$$T(n) = \sum_{i=0}^{\log n} k = \log n \; k$$

$$= O(\log N)$$

Fibonacci Series

$fib(n) = fib(n-1) + fib(n-2)$

0 1 1 2 3 5 8 ....

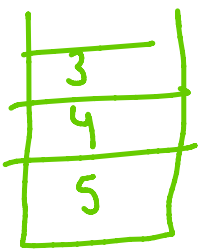


$1 \; 2 \; 4 \; 8 \; \vdots \; 2^{N-1}$

$2^0 \; 2^1 \; 2^2 \; 2^3$

$k(1 + 2 + 4 + 8 + \ldots + 2^{N-1})$

$= k \dfrac{(2^N - 1)}{2 - 1} = O(2^N)$



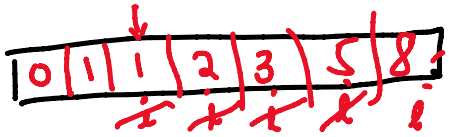
$k \frac{(a^n - 1)}{a-1} = O(a^n)$

a b b
a b c c

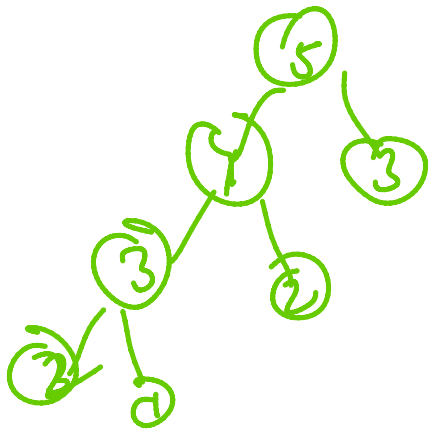
0 1 1 2 3 5 8 ...

3 approaches :

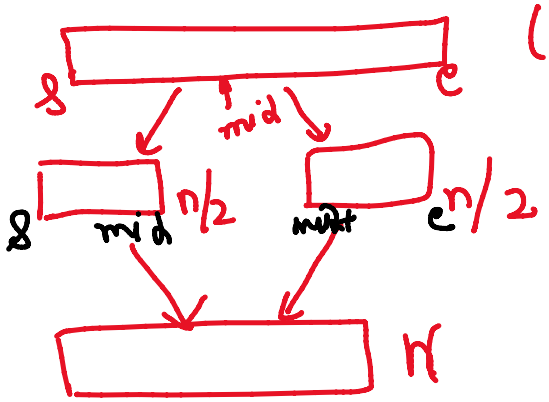
1. 3 variables. $O(N)$, $O(1)$
2. dp array $O(N)$, $O(N)$
3. Recursion $O(2^N)$, $O(N)$



$dp[i] = dp[i-1] + dp[i-2]$



Merge Sort $O(N \log N)$



(n) k

$2\,T(n/2)$

$k \cdot N$

$T(n) = K + 2T(n/2) + k \cdot N$

$T(n) = NK' + 2T(n/2)$

$2T(n/2) = \frac{N}{2}K' + 4T(n/4)$

$\ldots = NK' + 8T(n/8)$

$4T(n/4) = 4\frac{N}{4}k' + 8T(n/8)$

$\vdots$

$T(1) = k' + 0$

$$T(n) = \sum_{i=1}^{\log N} NK$$

$= N \log N \, K. = O(N \log N)$

Two Sum

25 September 2022 15:53