Plan - Sunday

Friday, 24 September 2021 5:32 PM

Plan for Fri, Sat, Sun

Day 1

- ✓ (i) Sieve of eratosthenes
- ✓ (2) Segmented sieve
- ✓ (3) Two diff.
- ✓ (4) Boats to save people
- (5) Min jump with +i, -i
- ✓ (6) Partition Labels

Day 2

- (i) Sliding window max
- (2) Max product Subarray
- (3) NXN → Transpose
- (4) N*M - Transpose
- (5) Rotate Image
- (6) Push Dominos

Day 3

- (i) Consecutive no. Sum
- (2) Add String
- (3) Multiply strings
- (4) Max Sum of two non overlapping subarray
- (5) Trapping rain water

Sieve Of Eratosthenes

Friday, 24 September 2021 6:13 PM

$N = 100$] → upper Range → upper limit → max allowed size of array!

assumption

$$\boxed{2 \le i_n \le N}$$

queries → $q_1 \to i_1$ +ve Integer

$q_2 \to i_2$

q_3

$\vdots \quad \vdots$

$q_{10,000} \quad i_{1000}$

i_n is prime or not

Given → $N < q$

⇒ check for every query if it is prime or not?

Allowed time complexity for all queries = $O(N+q)$

Given $q > N$.

$$\boxed{\text{Time} = O(q)}$$

, Space = $O(N)$

optimised Algo. for prime

⇒ Time = $\sqrt{N}$ for single Number.

time complexity of finding prime for 'q' queries

Time ⇒ $q \times \sqrt{N}$

$$\boxed{T = O(q\sqrt{N})}$$

observe → There is some kind of pre-calculation, which is helping us to solve the query in $O(1)$ time complexity.

N = 30 . Pre calculation → boolean array of size → n+1

max Range = N

Factor of all numbers are present till ≤ √n

2, 3, 5 →

~~0~~] Neither prime
~~1~~] nor composite

Multiple of 2 are not prime

2 - T Prime

3 - T Prime.

4 is already marked as not prime

4 - ~~T~~ F

5 - T prime.

6 - ~~T~~ F

fi Multiple are already marked

7 - T → prime

8 - ~~T~~ F

9 - ~~T~~ F

10 - ~~T~~ F

11 - T → prime

12 - ~~T~~ F

13 - T → prime

14 - ~~T~~ F

15 - ~~T~~ F

16 - ~~T~~ F

17 - T → prime

18 - ~~T~~ F

19 - T → prime.

20 - ~~T~~ F

21 - ~~T~~ F

22 - ~~T~~ F

23 - T → prime

24 - ~~T~~ F

25 - ~~T~~ F

26 - ~~T~~ F

27 - ~~T~~ F

28 - ~~T~~ F

29 - T → prime

30 - ~~T~~ F

inverse After Rootn

a x b = N, b x a = N

isprime [i] = False
↳ i is Not prime.

→ True
'i' is a prime No.

queries of upper Range till N → O(1) for single query

→ initially consider all as prime

Time complexity for pre-calculation → Iteration for $n = 2, 3, 5.$

$$\text{Time (generic)} = \underset{\substack{\uparrow \\ \text{for } 2}}{\frac{n}{2}} + \underset{\substack{\uparrow \\ \text{for } 3}}{\frac{n}{3}} + \underset{\substack{\uparrow \\ \text{for } 5}}{\frac{n}{5}} + \cdots\cdots + \underset{\substack{\uparrow \\ \text{for } K}}{\frac{n}{K}}$$

K is last prime number b/w 0 to $\sqrt{N}$

$$= n\underbrace{\left[\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \cdots\right]}_{\text{Sum of inverse of prime Number,}}$$

Math → converge - divergence (cocept to solve this equation) → Common

$$\log_{a} a^{n} = n$$

$$\log_2 \log_2 (2^{32}) \quad \log_2(32) = \log_2(2^5) = 5$$

$$\Downarrow \qquad \downarrow \qquad 10^9$$

└→ Result = $\log(\log(n))$

Max value for integer.

$$\text{overall time complexity} = O[n \log\log(n)] \xrightarrow{\text{equivalent}} O(n)$$

max → $\overset{\shortparallel}{5n}$

At max $5n$.

Segmented Sieve

Friday, 24 September 2021 7:30 PM

Range of numbers $[a, b]$ Both including

$N = b - a$

Constraints

(i) $1 \le a \le b \le 10^9$

(2) $b - a \le 10^5$

(1) given some queries,

(2) Check for i^{th} query that query[i] is prime or not?

Approach. (1) Travel from a to b and check if every No. is prime or Not.

Time → $N \times \sqrt{b}$

Approach 2 → (Normal) Sieve Algorithm → Doesn't work. why?

upper Range is of order 10^9

Array creation is not possible.

Approach 3 - optimised sieve Algo (Segmented Sieve Algo)

→ $(b + q) \Rightarrow$

on small scale, a = 22, b = $\boxed{50}$

→ $b \le 10^9$ = $\sqrt{b}$ = 10^5

upperlog Segmented array. , $b - a \le 10^5$

starting index of prime[i]

diff = b - a = 50 - 22 = 28 Prime till Root(b) → [2, 3, 5, 7] sieve

array cam be created.

Index-val By default

0 - 22 → ~~F~~ T

1 - 23 → F → prime

2 - 24 → ~~F~~ T

3 - 25 → ~~F~~ T

4 - 26 → ~~F~~ T

5 - 27 - ~~F~~ T

$\boxed{6 - 28}$ - ~~F~~ T

7 - 29 - F → prime

Index-val

8 - 30 - ~~F~~ T

9 - 31 - F → prime

10 - 32 - ~~F~~ T

11 - 33 - ~~F~~ T

12 - 34 - ~~F~~ T

13 - 35 - ~~F~~ T

14 - 36 - ~~F~~ T

15 - 37 - F → prime

Index-val

16 - 38 - ~~F~~ T

17 - 39 - ~~F~~ T

18 - 40 - ~~F~~ T

19 - 41 - F → prime

20 - 42 - ~~F~~ T

21 - 43 - F → prime

22 - 44 - ~~F~~ T

23 - 45 - ~~F~~ T

Index-val

24 - 46 - ~~F~~ T

25 - 47 - F → prime.

26 - 48 - ~~F~~ T

27 - 49 ~~F~~ T

28 - 50 ~~F~~ T

if isprime[i] is false then value associated with index is prime.

How to find find Index of Multiple of prime[i] →

a = 22 Smaller

b = 50

prime[i] = 7

7, 14, 21

$$\text{multiple} = \left\lceil \frac{a}{\text{prime}[i]} \right\rceil_{\text{ceil}} = \left\lceil \frac{22}{7} \right\rceil_{\text{ceil}} = \lceil 3.1\ldots \rceil_{\text{ceil}} = \boxed{4}$$

$$\text{First Multiple} = \underbrace{\text{prime}[i]} * \underbrace{\text{multiple}} = 7 * 4 = \underline{28}$$

$$\text{Index of First Multiple} = \text{First Multiple} - a = 28 - 22 = \boxed{6} \text{ Index}$$

a ↑ Base Value

if multiple is equal to 'L' that mean first value of Range is prime itself → start with (+1) multiple (i.e. next multiple)

Range = 7 to 50, prime[i] = 7

$$\text{multiple} = \left\lceil \frac{a}{\text{prime}[i]} \right\rceil_{\text{ceil}} = \left\lceil \frac{7}{7} \right\rceil_{\text{ceil}} = \boxed{1} = \text{multiple}$$

prime no. is marked

$$\text{First Mult} = 7 * \cancel{1}\ 2 = \boxed{7}\ \boxed{14}$$

firstmult − a

$$\text{Index} = 7 - 7 = \boxed{0}$$

$$14 - 7 = \boxed{7}$$

0	~~7~~	T
1	8	
2	9	
3	10	
4	11	
5	12	
6	13	
7	~~14~~	T
8	15	
9	16	

Conclusion:

To convert 'a' into double

```
→ int multiple = (int) Math.ceil( a * 1.0 / prime[i] );

→ if (multiple == 1) multiple ++;

→ int Starting Index = multiple * prime - a;

for(int j = si ; j < isprime.length ; j += prime[i]) {
   // mark isprime[j] = True;
}
```

→ Traval and check if it is prime or not

$\frac{int}{int} = int$, $\frac{5}{3} = 1.67$

ceil = 2

Edge case

[(1) , n]

↑

Included

→ management of 1

Pair With Given Difference

Friday, 24 September 2021 9:04 PM

target diff = 78

~~No. of pair,~~ availability of pair having difference equal to target.

arr → 82 5 20 3 2 5 80 81

Sort → 2 3 5 5 20 80 81 83

left

right

target sum => left = 0

right = n-1

NOTE: if left and right point at opposite end then redundancy will encounter.

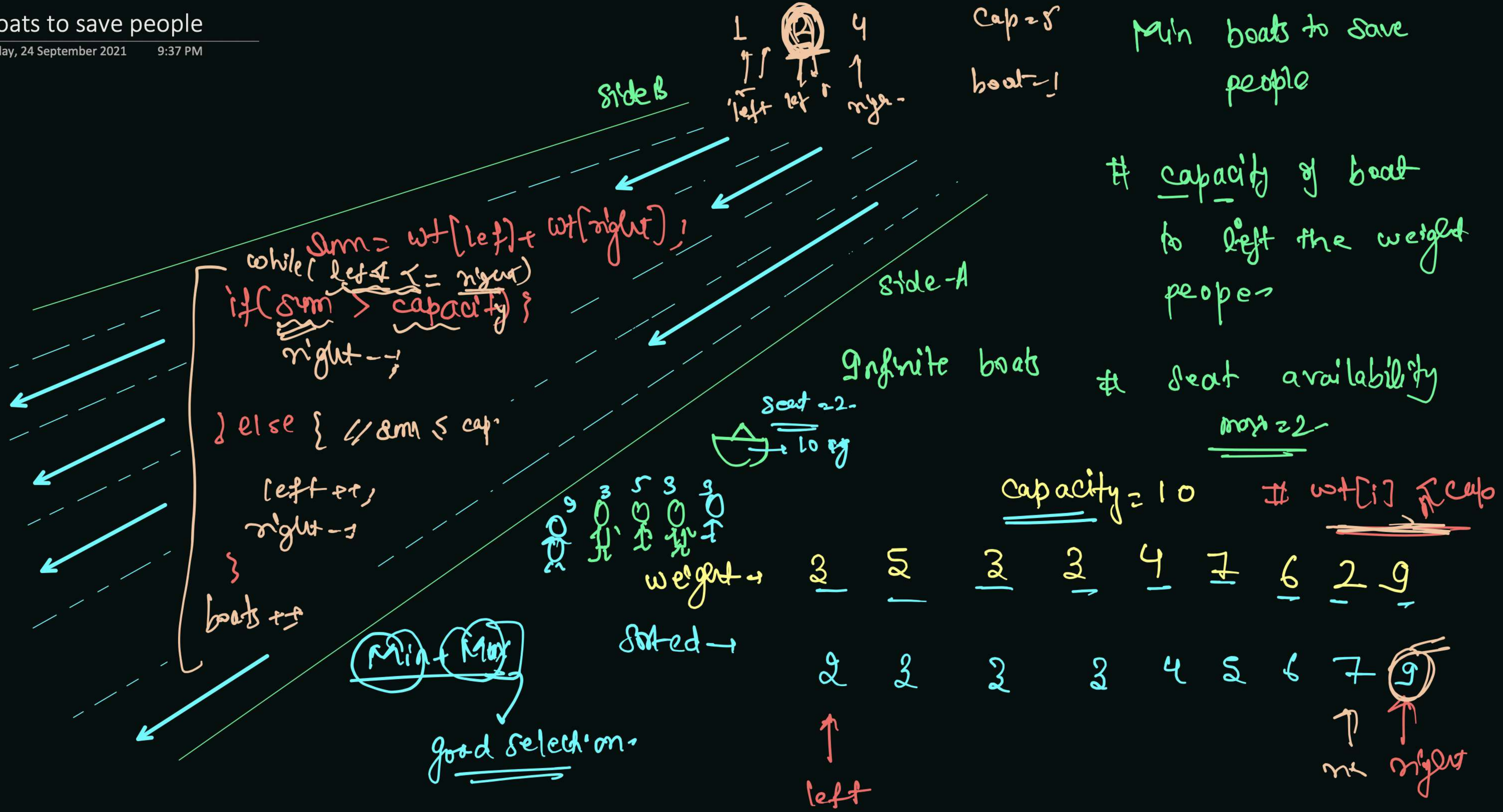
diff = ~~1~~ ~~2~~ 78

```
diff = arr[right] - arr[left]
if (diff == target) {
    return true;
} else if (diff > target) {
    left ++;

} else { // diff < target
    right ++
}
```

Boats to save people

Friday, 24 September 2021 9:37 PM



Cap = 8

boat = 1

Min boats to save people

capacity of boat to lift the weight peopes

seat availability

max = 2

Infinite boats

```
Sum = wt[left] + wt[right];
while( left <= right)
if(sum > capacity) {
    right--;

} else { // sum <= cap

    left++;
    right--;
}
boats++
```

Min + Max

good selection.

capacity = 10

wt[i] <= cap

weight → 2 5 3 3 4 7 6 2 9

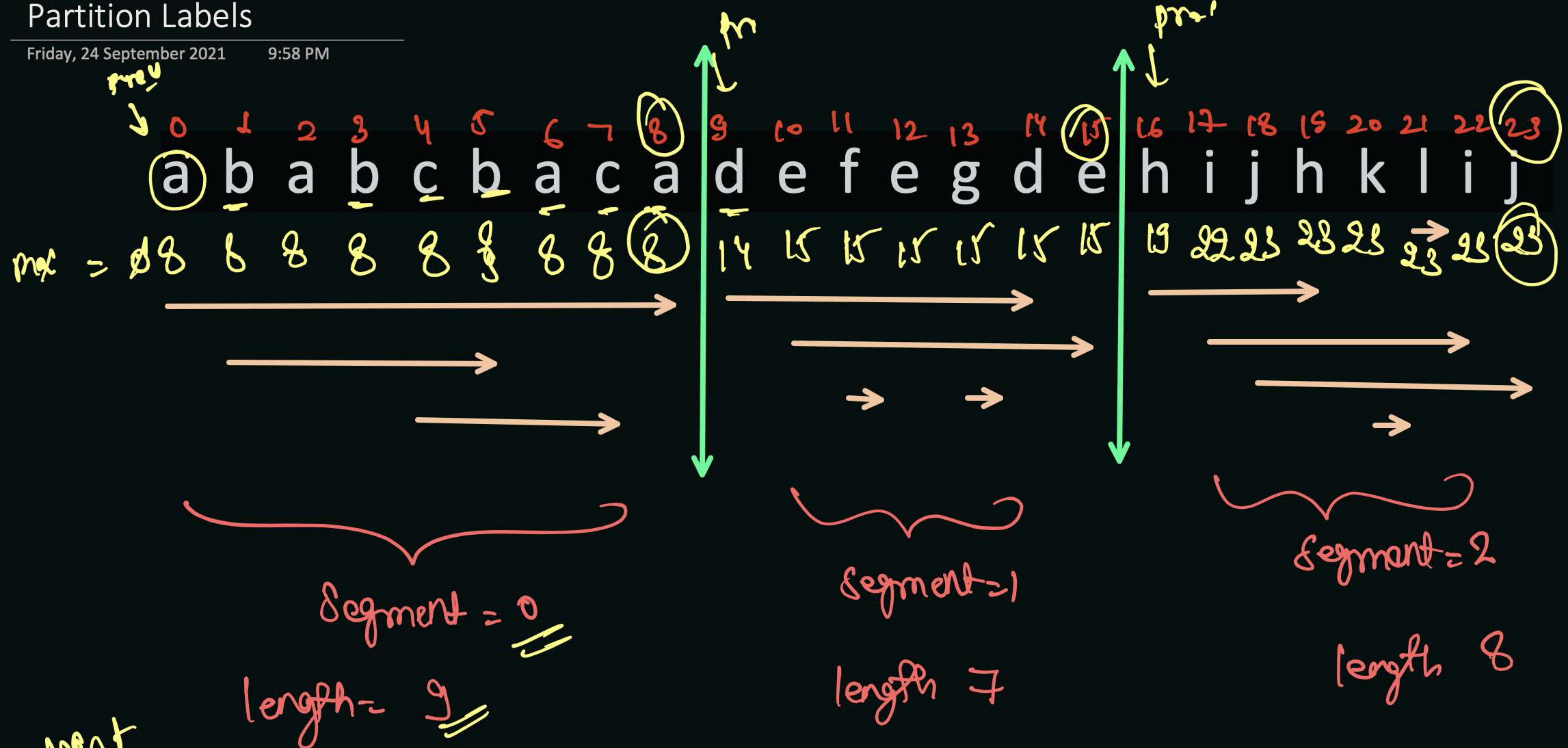
sorted → 2 2 2 3 4 5 6 7 9

left

right

Partition Labels

Friday, 24 September 2021 9:58 PM



⇒ max Impact using chaining technique

Hint

Similar problem → chunks to make to make array sorted 1,2

Hashm→ O(26) → constant

Hint

HashMap → last occurence Index

character vs. Integer

array of size 26

Result = [9, 7, 8]

time - O(n)

space - constant space