Maximize distance to closest one

Saturday, 18 September 2021 2:23 PM

1 → occupied chair

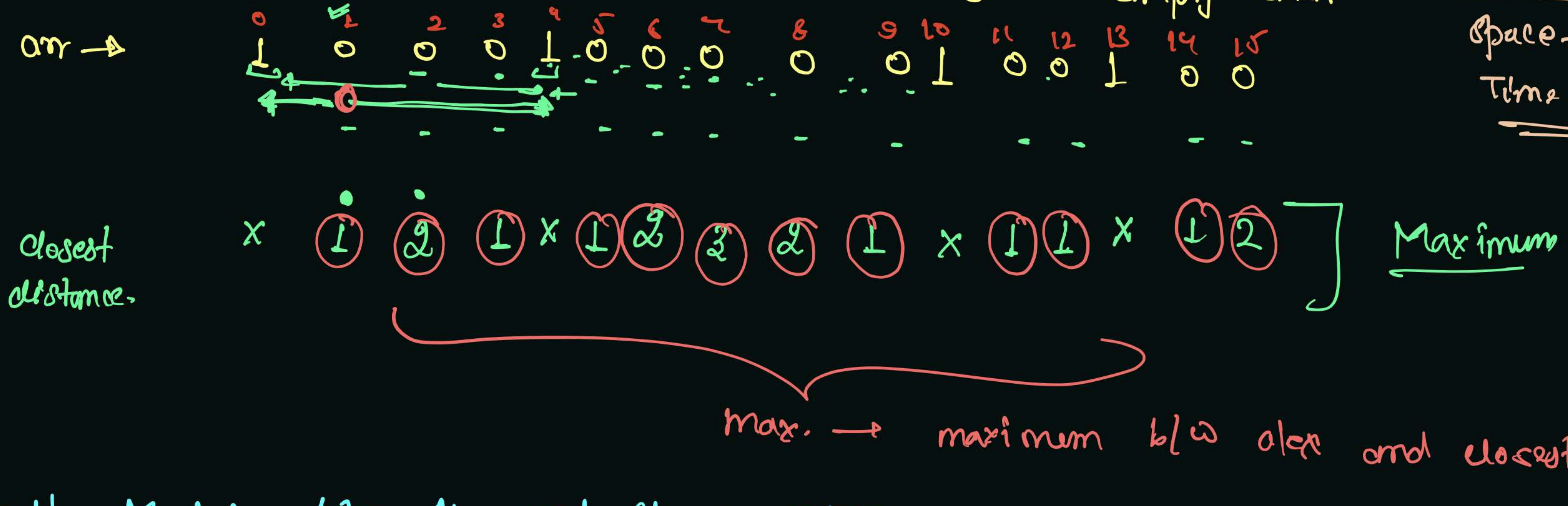
0 → empty chair

Approach-0

Space → O(n)

Time → O(n)

max. distance b/w closest person & Alex



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
arr →	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0
closest distance.	x	1	2	1	x	1	2	3	2	1	x	1	1	x	1	2

Maximum

max. → maximum b/w alex and closest person

Result = Maximise b/w Alex and Closest person.

closest person = Min b/w left person to Alex and Right person to Alex.

arr →	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0
left person distance.	∞	∞	∞	∞	0	1	2	0	1	2	3	4	5	0	1	2	0	1	2
right person distance	4	3	2	1	0	2	1	0	5	4	3	2	1	0	2	1	0	∞	∞
closest distance min	4	3	2	1	0	1	1	0	1	2	3	2	1	0	1	1	0	1	2

max Res

Indx ~~∞~~ ~~2~~ ~~7~~ 13

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
|∞|	|∞|	-1	1	2	3	4	-1	1	2	3	4	5	-1	1	2	3	4
2	1	x	4	3	2	1	x	5	4	3	2	1	x	∞	∞	∞	∞
2	1	-1	1	2	2	1	-1	1	2	3	2	1	-1	1	2	3	4

2 ~~7~~ ~~13~~

Indx = ~~∞~~

virtual closest b/w index vs right

closest → $arr[i] = Math.min(4, Indx - i)$

to maximise closest distance → $max = Math.max(max, arr[i])$

$max = 4$

steps-

① Iterate on array and make left persons distance on given array Except for occupied seat.

② Iterator from Right to left and manage closest distance as well as max distance b/w closest and also

First Missing Positive

Saturday, 18 September 2021 2:23 PM

Example →	4	2	7	-1	3	8	12	1	-2
Index	0	1	2	3	4	5	6	7	8

First Missing +ve ??

Approach 2 -

① Fill All Elements in Hashmap.

② Try to find presence from 1 to n, if number is not present, return that number.

otherwise → return n+1] last positive number

Time - O(n)

Space - O(n)

Approach - 0

n = length

Try to search from 1 to n

if absent → return that num

otherwise → return n+1

Time - $O(n^2)$

Space - O(1)

Approach 1

① Sort array

② Try to find first +ve and then check its continuity.

if not continuous return next continuous number

otherwise → return last +ve +1

Time - O(n logn)

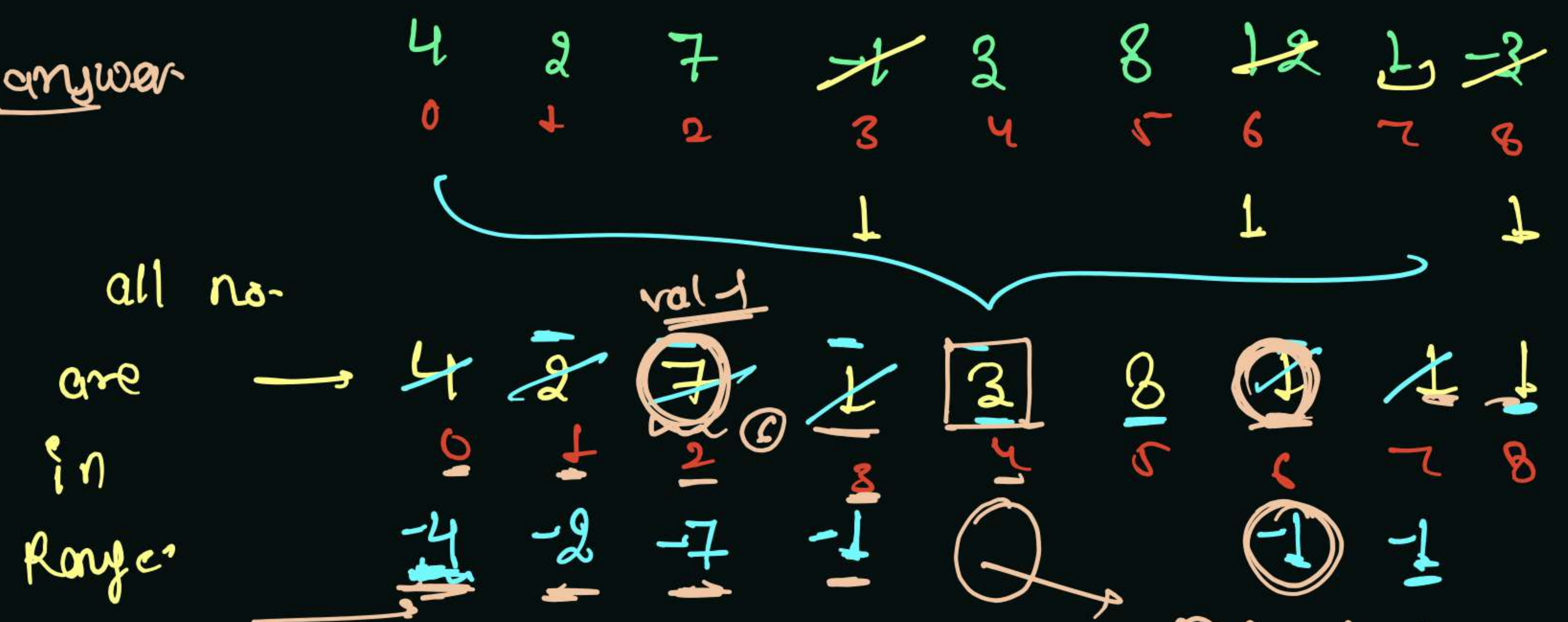
Space - O(1)

Exomple →	4	2	7	-1	3	8	12	1	-2
Index	0	1	2	3	4	5	6	7	8

Optimised Approach: →

If it conteins 1 to n, n+1 is anywer

num ≤ 0 || 9 < num.



Step1. Travel and mark no. which are out of Range as 1

$\text{num} \le 0 \;||\; n < \text{num}$

out of Range.

boolean oneIsPresent = ~~False~~; True.

5 is missing i.e. why 4th index is stay in +ve

NOTE: Simultaneously check the presence of ones

If ~~marker~~ is +ve → convert into -ve

if in -ve → leave as it is.

Step2 → If one is absent → return 1!
otherwise → solve using mangement of Index.

Step3 → travel and find unmark Index, Index + 1 is Result otherwise n+1

Best Meeting Point

Saturday, 18 September 2021 2:23 PM

Village

Steps of Discussion →

① What?

② Why?

③ How?

P.

P_1	1	0	0	0	1	P_2
	0	0	0	0	0	
	0	0	1	0	0	

P_3

Meet at common point.

Such that combination of travel of all person will min,

distan = $|P - P_1| + |P - P_2| + |P - P_3|$

distance ⟶ minimum ↑

How to calculate distance

→ using Manhattan distance formula →

$P_1 \longrightarrow P_2$

(x_1, y_1) (x_2, y_2)

distemce = $\left.|P_1.x - P_2.x| + |P_1.y - P_2.y|\right]$ distance.

What → we have person $\underbrace{P_1, P_2, P_3 . P_4}_{\text{coordinate of person}}$ ---- in a village.

✓ x- coordinate → x_2, x_2, x_3, x_4, x_5 ---

No. of person = 5

Sort x-coordinate = After sorting = $x_1, x_2, \boxed{x_3}, x_4, x_5$

median

✓ y- coordinate → y_1, y_2, y_3, y_4 -----

median

Sort y- coordinate ⇒ After sorting = $y_1, y_2, \boxed{y_3}, y_4, y_5$

Best Meeting point ⇒ Median from x coordinate, mediane from y- coordinate

$|P_1 - P| = |P_1x - P.x| + |P_1.y - P.y|$

⇒ $P \Rightarrow (x_3, y_3)$ → Best Meeting point

find distance of all person from meeting point → distance = $\underbrace{|P_1 - P|} + |P_2 - P| + |P_3 - P| + |P_4 - P| + |P_5 - P|$

$$\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 \end{array}$$

x-coordinate → 0, 0, 0, $\boxed{1}$, 1, 2, 2

How to get x-coordinate in sorted order
→ travel row wise

y-coordinate → 0, 0, 1, $\boxed{2}$, 2, 4, 4

Column wise traversal to get sorted y.

$P_1 \to (0,0)$

$P_2 \to (0,1)$

$P_3 \to (0,4)$

$P_4 \to (1,0)$

$P_5 \to (1,2)$

$P_6 \to (2,2)$

$P_7 \to (2,4)$

Median = $\underset{x}{1}$, $\underset{y}{2}$

Meeting point → (1, 2)

$P = (1,2)$

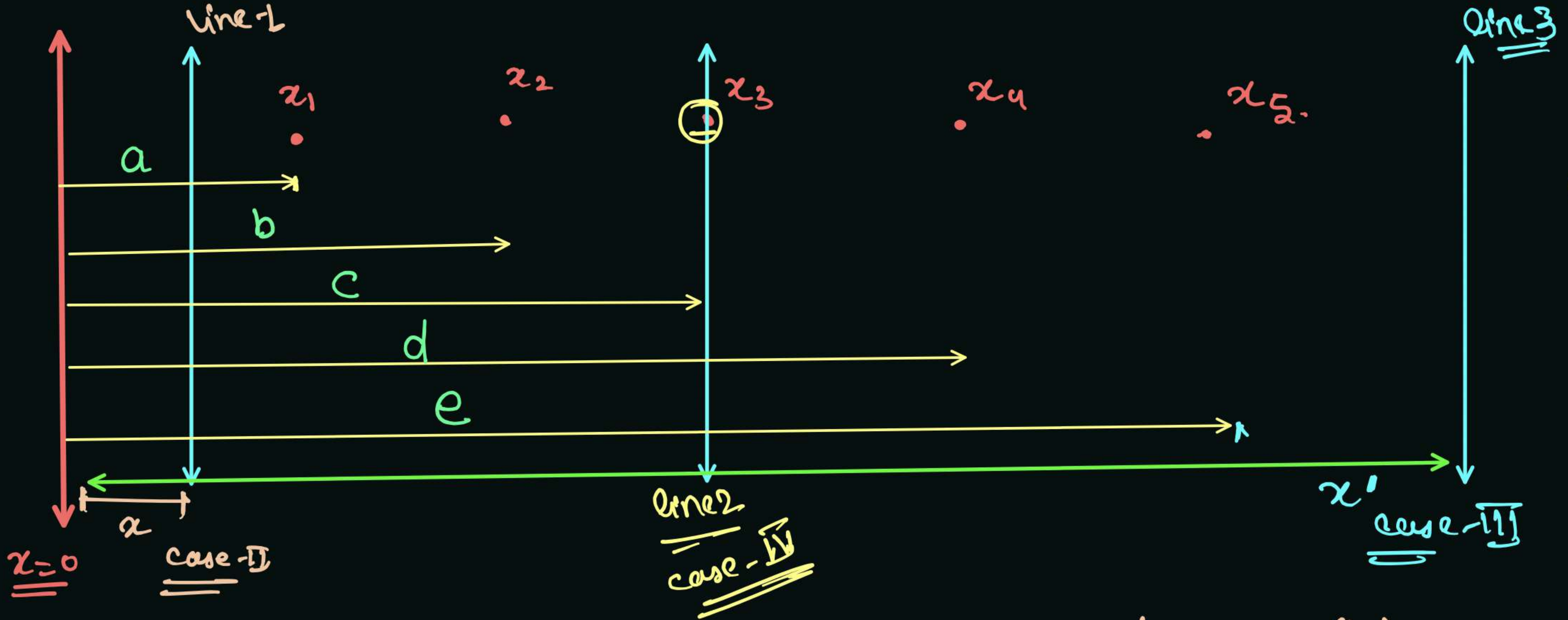
$$\text{find dist} = P_1 \leftrightarrow P + P_2 \leftrightarrow P + P_3 \leftrightarrow P + P_4 \leftrightarrow P + P_5 \leftrightarrow P + P_6 \leftrightarrow P + P_7 \leftrightarrow P$$

$$= \underbrace{|0-1|}_{x} + \underbrace{|0-2|}_{y} + \underbrace{|0-1|}_{x} + \underbrace{|1-2|}_{y} + \underbrace{|0-1|}_{x} + \underbrace{|4-2|}_{y} + \underbrace{|1-1|}_{x} + \underbrace{|0-2|}_{y} + \underbrace{|1-1|}_{x} + \underbrace{|2-2|}_{y}$$

$$+ \underbrace{|2-1|}_{x} + \underbrace{|2-2|}_{y} + \underbrace{|2-1|}_{x} + \underbrace{|4-2|}_{y}$$

$$= 1 + 2 + 1 + 1 + 1 + 2 + 0 + 2 + 0 + 0 + 1 + 0 + 1 + 2$$

$$= \boxed{19} \text{ Ans}$$



Case-I

Meeting point is at $x=0$

distance = $x_1+x_2+x_3+x_4+x_5$

di = $a+b+c+d+e$

$$\boxed{\text{dist} = a+\underbrace{b+c+d}_{}+e}$$

d1

Case-II (Moving toward Right)

If meeting point is line-1

distance = $a-x+b-x+c-x+d-x+e-x$

$$\boxed{\text{dist} = a+b+c+d+e-5x}$$

d2

d2 is better than d1

case-IV (Meeting point is at medium)

distance = $\underline{x_1}+x_2+x_3+x_4+x_5$

dist = $\cancel{c}-a+\cancel{c}-b+0+d-\cancel{c}+e-\cancel{c}$

$$\boxed{\text{dist} = d+e-(a+b)}$$

medium have less distance traversal count.

case-III (Extreme Right)

If meeting point in line 3

distance = $x'-e+x'-d$ $+x'-c+x'-b+x'-a$

$$= 5x'-\underbrace{(a+b+c+d+e)}$$

$\underline{5x'} > 5*(a+b+c+d+e)$

$$\boxed{\text{dist} = 5x'-(a+b+c+d+e)}$$

d3 is equivalent to dist 1

Maximum Swap

Saturday, 18 September 2021 2:23 PM

maximum one swapping is allowed,

→ max possible number.

✗ LSD → swapping.

✓ MSD → swapping max possible

	0	1	2	3	4	5	6	7	8	9
String num =	9	9	8	~~7~~	5	6	7	3	~~8~~	4
				8					7	

9 to

Right max

	0	1	2	3	4	5	6	7	8	9
				8					7	
num =	9	9	8	~~7~~	5	6	7	3	~~8~~	4
→	1	1	8	8	8	8	8	8	8	9

is it necessary?

✗ right max array

" 9 9 8 8 5 6 7 3 7 4 " → Res

10-size space last occurr. (Digit array)

9 →	~~0~~ 1
8 →	~~2~~ 8
7 →	~~3~~ 6
6 →	5
5 →	4
4 →	9
3 →	7
2 →	
1 →	
0 →	

9 to current digit

Exclusive

9
8
7
6
5