

Morning

- ① Min jumps with $+i$ $-i$ moves
- ② Transpose $N \times N$
- ③ Transpose $M \times N$
- ④ Rotate Image
- ⑤ Partition Labels
- ⑥ Add String.

Evening

- ① Buddy Nim
- ② Min no. of Boats
- ③ Min. Platforms
- ④ max. product subarray
- ⑤ Multiply String

Min Jumps with +1 -1 move

Sunday, 12 September 2021

10:39 AM

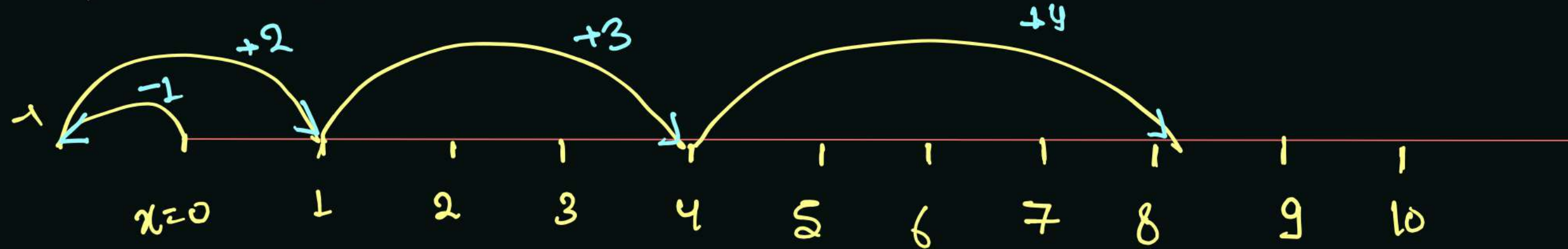


Min → Reach to

nearest point of target.

$$x=8$$

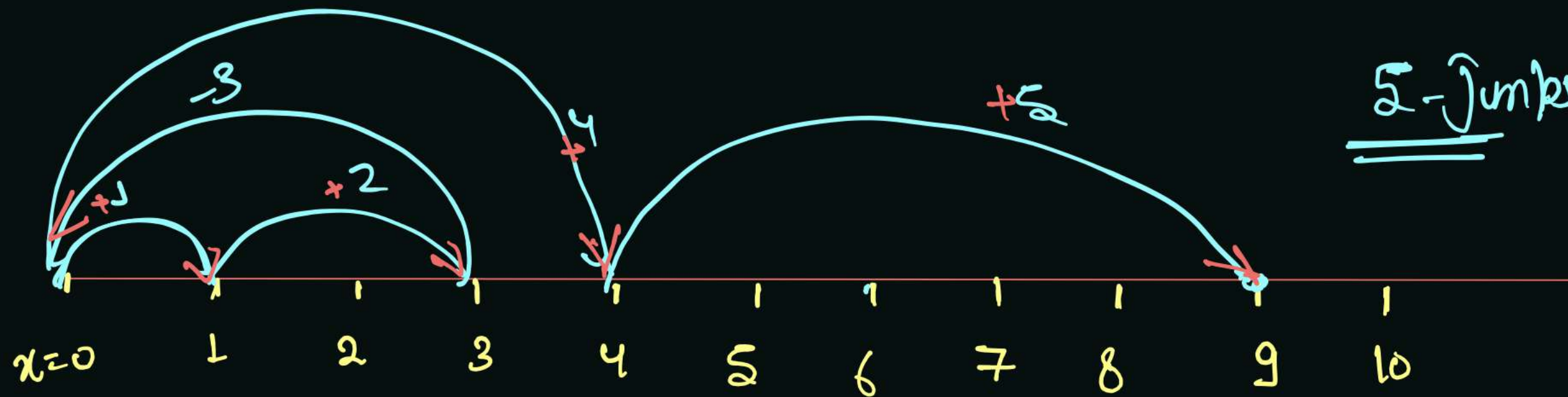
4-jumps → 8



jump → successive

$$\begin{aligned} &(-1, +1) \text{ (1)} \\ &(-2, +2) \text{ (2)} \\ &(-3, +3) \text{ (3)} \end{aligned} \quad \underline{\underline{+1, -1}}$$

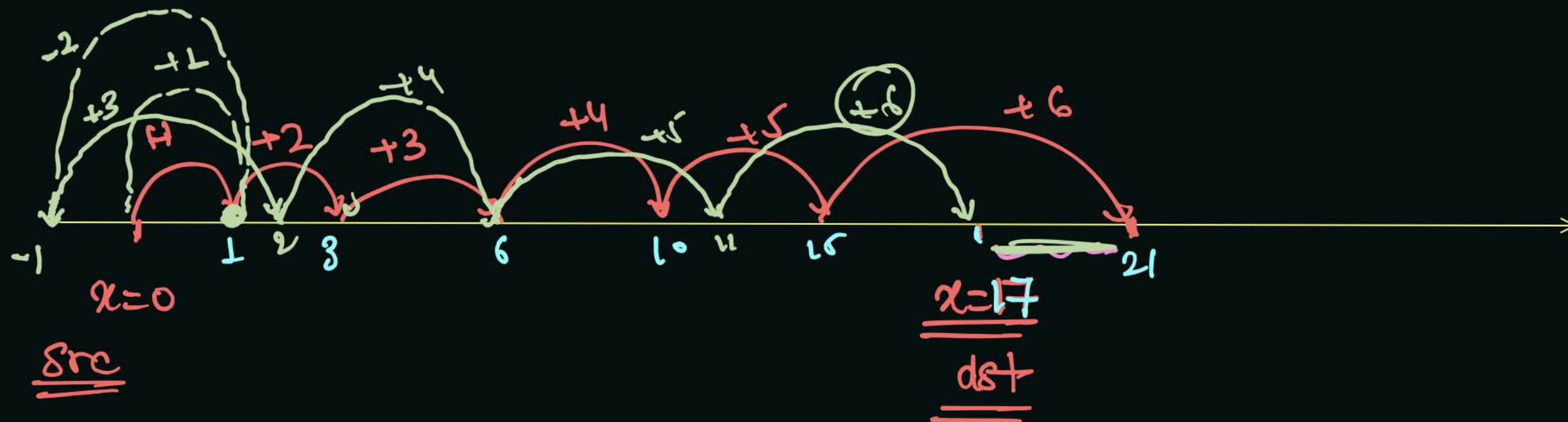
Skip of jump is
not allowed



5-jumps to reach at

✓ # If we are taking jump toward target point, then it is helpful to ensure min jumps

But it is not necessary that we can reach at destination using +ve jumps



If we will reach at destination then jump is answer.

If we will the destination point

$$21 - 17 = 4$$

diff. even

make half jump

count

as -ve

is final Result

↑ no. of jumps

diff odd.

-ve direct

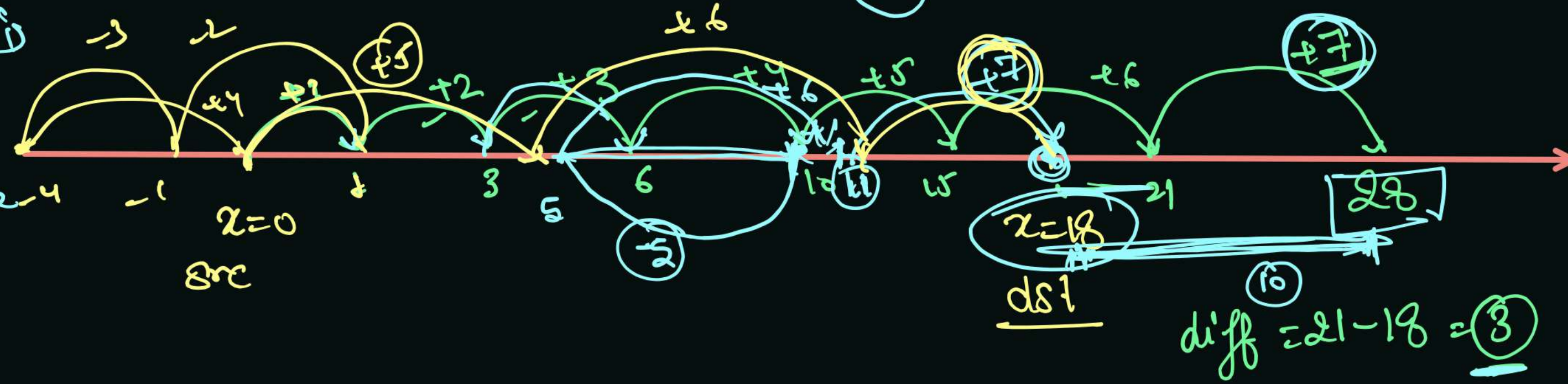
$$21 - 17 = 4$$

$$\text{half} = \frac{4}{2} = 2$$

if diff after crossing destination is odd.

no. of jumps

if target is
we take
absolute
value



$$\frac{10}{2} = 5 \text{ (ve)}$$

Sum of jump

$$\text{diff} = 10$$

$$\frac{\text{diff}}{2} = -ve$$

$$-\frac{10}{2} = -5$$

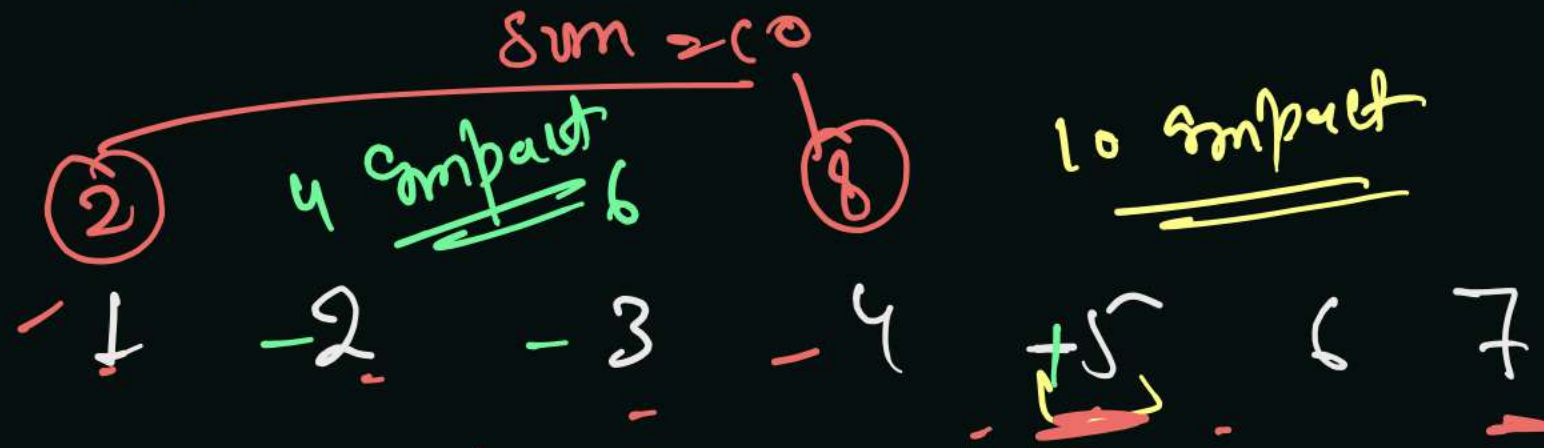
if diff is odd, take another next jump

Even → Count is
Result after
jump
Odd

if odd diff encounter then take one more

jump and it is our final result because

next difference is Guaranteed Even. 1+2+3

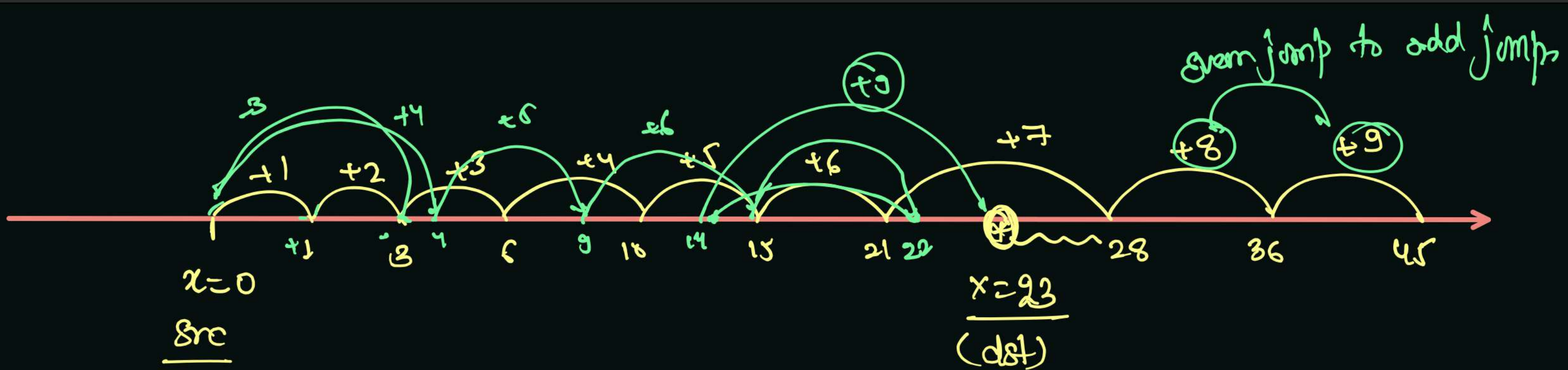


Jump count remain same

$$\text{Sum} = 28 + 10 = 18 \text{ dest}$$

$$28 - 4 - 6 = 18$$

$$28 - 2 - 2 = 24$$



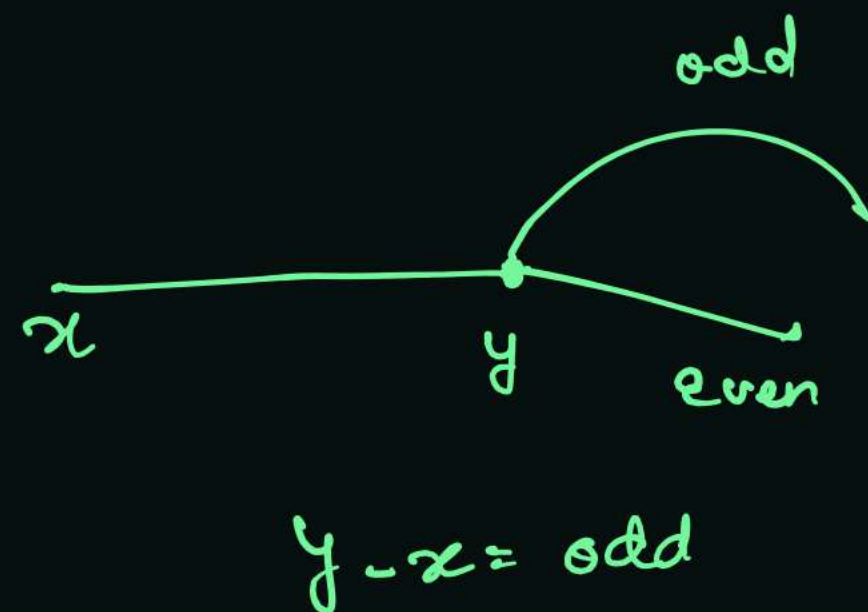
$$\text{diff} = 28 - 23 = \underline{5}$$

diff 1 = odd \rightarrow jump one more

$$\text{diff} = 28 - 23 = 5$$

$$\hookrightarrow \text{diff } 2 = 36 - 23 = 13$$

$$\hookrightarrow \text{diff } 3 = 45 - 23 = 22$$



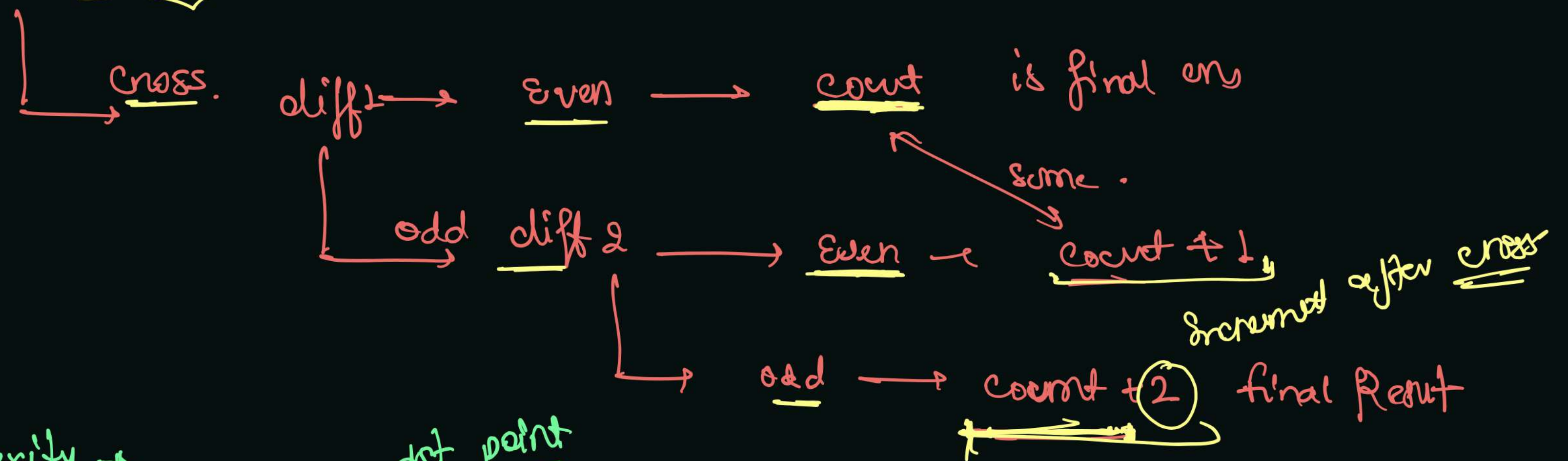
$$\begin{aligned} &\text{odd} + \text{even} + \text{odd} = \text{even} \\ &y-x = \text{odd} \quad \text{Even} \Rightarrow \text{diff} = \text{odd} \\ &y-x \rightarrow \text{odd} + \text{odd} \Rightarrow \text{diff} = \text{Even} \end{aligned}$$

no. of ways can be vary but jump count remain same

$$\begin{aligned} \frac{\text{diff}}{2} &= \frac{22}{2} = 11 \quad \text{half unit} \\ &= 11 \\ &= 11 \\ &= 11 \end{aligned}$$

5 6

if reach at destination \rightarrow count is final ans.



Time complexity \Rightarrow

dot point

$$1 + 2 + 3 + 4 + \dots \leq N$$

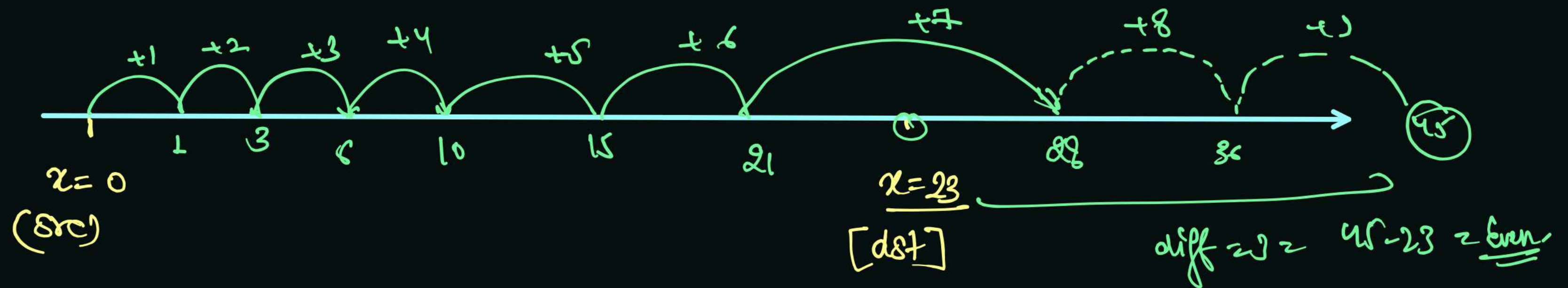
n terms

no. of jumps $\rightarrow \frac{n(n+1)}{2} = N$

$$n^2 \equiv N$$
$$\Rightarrow n \equiv \sqrt{N}$$

time complexity = $\mathcal{O}(\sqrt{N})$

where N is destination point



target = 23

jump = ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ **7**

s = ~~0~~ ~~2~~ ~~6~~ ~~10~~ ~~15~~ ~~21~~ **28**

diff 1 = $28 - 23 = 5$

→ Try for 7

diff 2 = $28 + 8 - 23 = 36 - 23 = 13$

→ Try for 8

diff 3 = Guaranteed

Even

→ Try for **9**

```

public int reachNumber(int target) {
    target = Math.abs(target);

    int jump = 0;
    int s = 0;

    while(s < target) {
        jump++;
        s += jump;
    }

    if(s == target) {
        return jump;
    } else if((s - target) % 2 == 0) {
        return jump;
    } else if((s + jump + 1 - target) % 2 == 0) {
        return jump + 1;
    } else {
        return jump + 2;
    }
}

```

diff 1 →

diff 2 →

diff 3 →

Transpose

Sunday, 12 September 2021

12:46 PM

$M \times N$

matrix \rightarrow

	0	1	2	3
0	A 0 0	B 0 1	C 0 2	D 0 3
1	E 1 0	F 1 1	G 1 2	H 1 3
2	I 2 0	J 2 1	K 2 2	L 2 3

Dimension = rows \times column
 $= 3 \times 4 = m \times n$

coordinate of element = (i, j)

transpose

A 0 0	E 0 1	I 0 2
B 1 0	F 1 1	J 1 2
C 2 0	G 2 1	K 2 2
D 3 0	H 3 1	L 3 2

resultant
dimension = $n \times m$

same element have
coordinate (j, i)

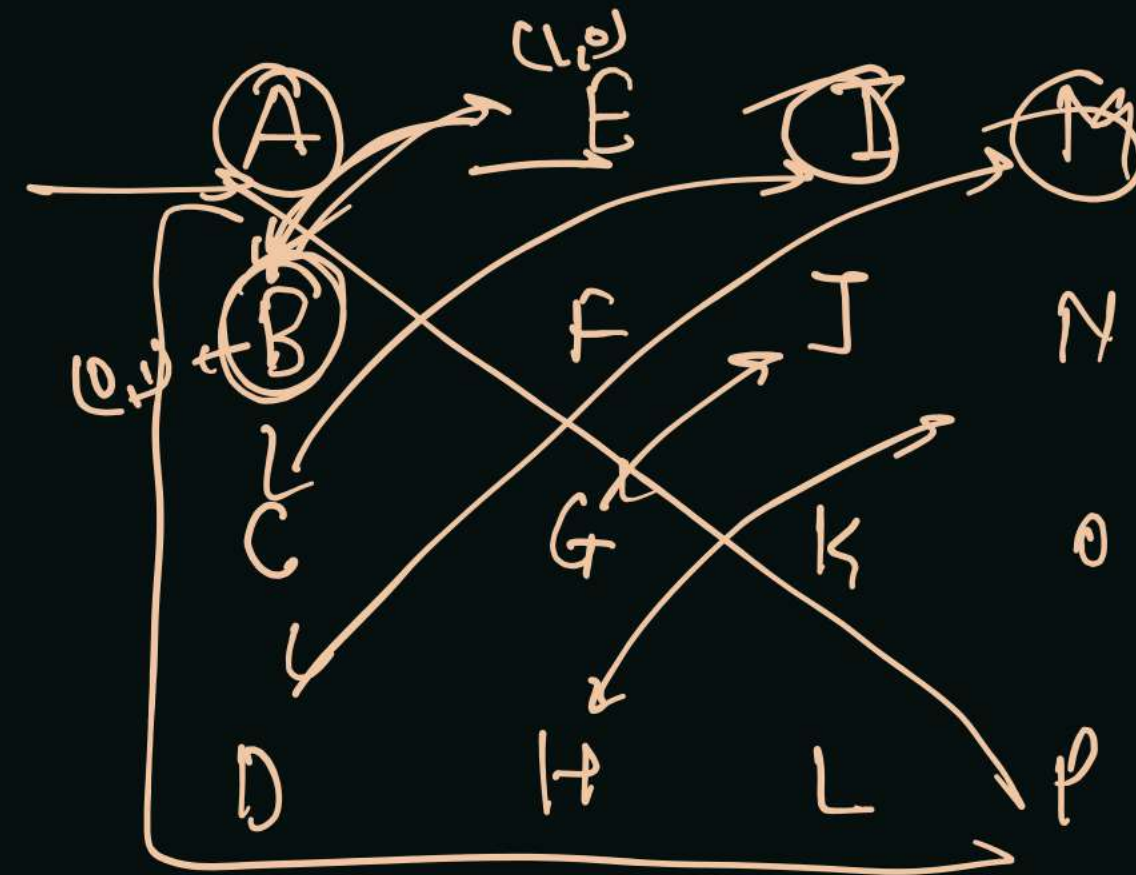
A 0 0	E 0 1	I 0 2
B 1 0	F 1 1	J 1 2
C 2 0	G 2 1	K 2 2
D 3 0	H 3 1	L 3 2

NxN - Transpose

	0	1	2	3
0	A	B	C	D
1	E	F	G	H
2	I	J	K	L
3	M	N	O	P

Transpose

$n \times n = 4 \times 4$



travel in half of matrix

result have dimension $n \times n$
ie. \rightarrow 4×4

matrix[i]

\rightarrow swap(matrix[i], j)

temp = matrix[i][j];

matrix[i][j] = matrix[j][i]

matrix[j][i] = temp;

Rotate Image

Sunday, 12 September 2021

1:01 PM

row 0 → column n-1

row 1 →

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Rotate 90°

Rotate 90° deg-
Clockwise

M	I	E	A
N	J	F	B
O	K	G	C
P	L	H	D

row and
column
swap
during
transp

Transpose

Steps:

- ① Transpose
- ② Reverse Every Row

A	E	I	M
B	F	J	N
C	G	K	O
D	H	L	P

reverse
M I E A

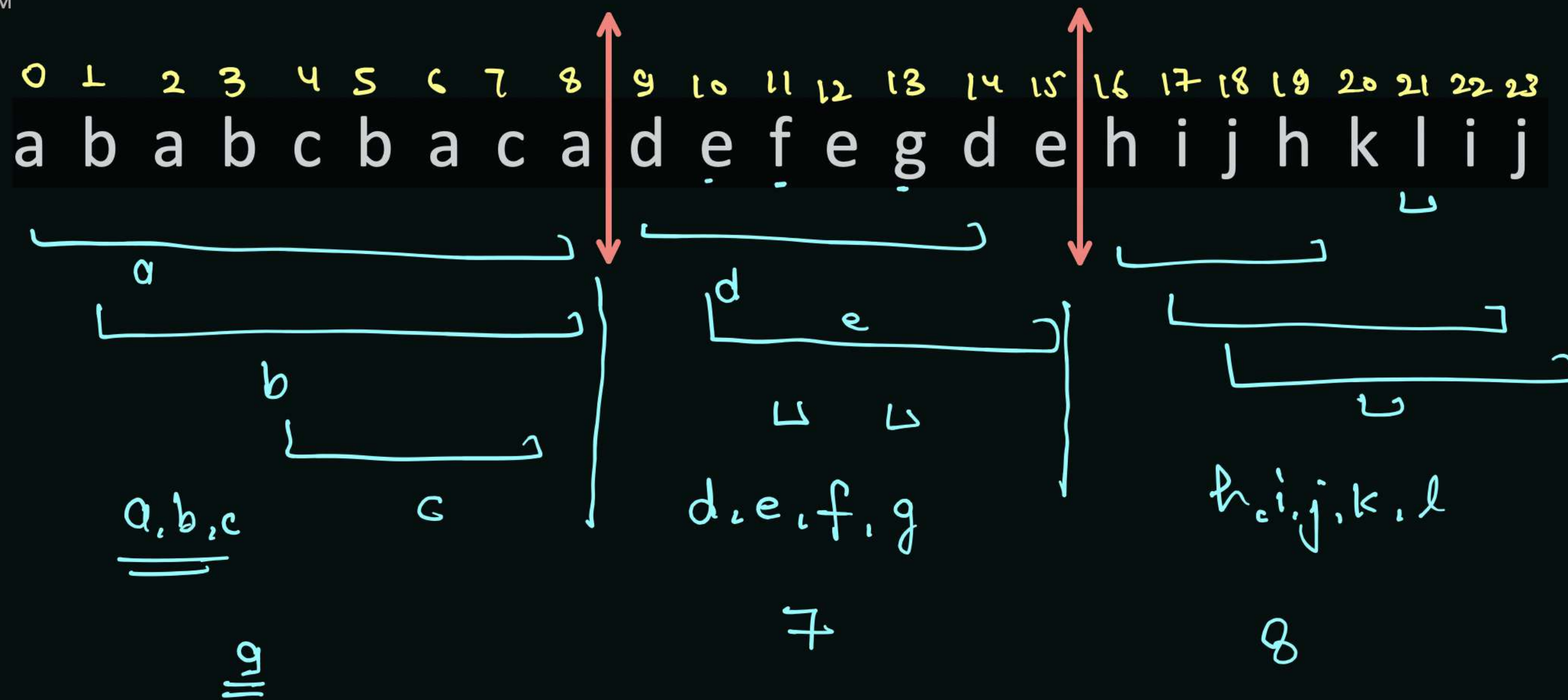
reverse Every row

Row and column swapping

Partition Labels

Sunday, 12 September 2021

1:10 PM



[9, 7, 8] result

Max No. of chunks = size.

Min no. of chunks = 1

Begin from 1:40

