Q) You have an array of all zeroes initially of length N. You need to perform Q queries on the array, of 3 types:

$N \leq 5 \times 10^5$
$Q \leq 5 \times 10^5$

1) X → update arr[x] to 2*arr[x]+1

2) X → Update arr[x] to $\lfloor arr[x]/2 \rfloor$

3) X Y → $\forall\ i \in [x, y]$ take all arr[i] & convert them into binary strings. Now concatenate all the strings and return the no. of set bits.

[0, 0, 0, 0, 0]

↓

1 x=1 [1, 0, 0, 0, 0]

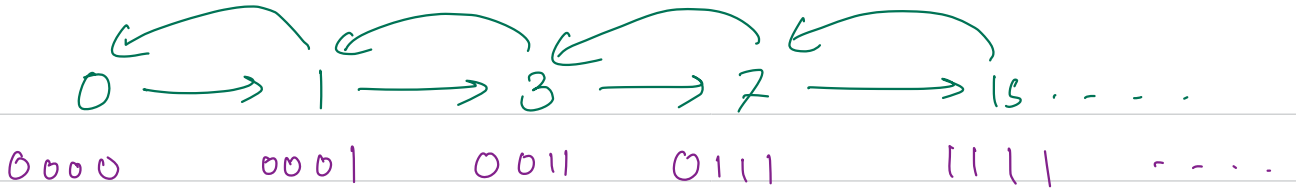
1 x=2 [1, 1, 0, 0, 0]

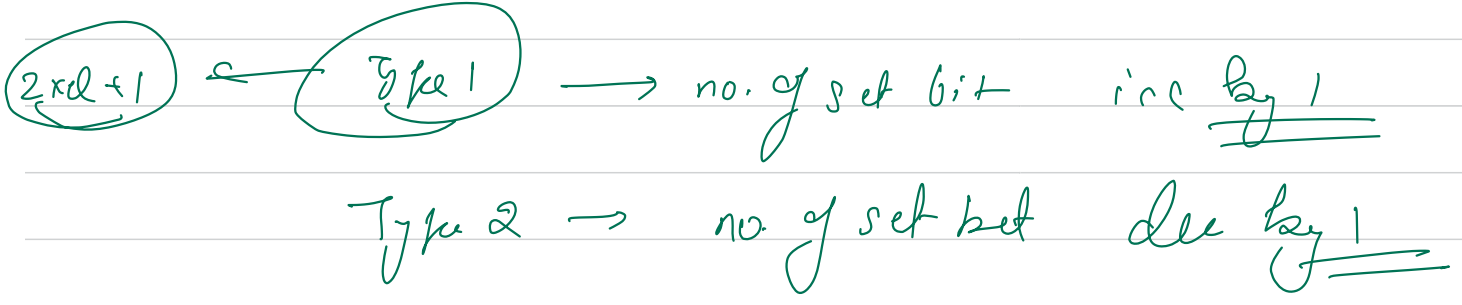
1 x3 [1, 1, 1, 0, 0]

3 1 3 ⟶ 3

3 2 4 ⟶ 2

→ print for type 3 only


0 → 1 → 3 → 7 → 15 . . . .
0000 0001 0011 0111 1111 . . . .


2*d+1
Type 1 → no. of set bit inc by 1
Type 2 → no of set bit dec by 1

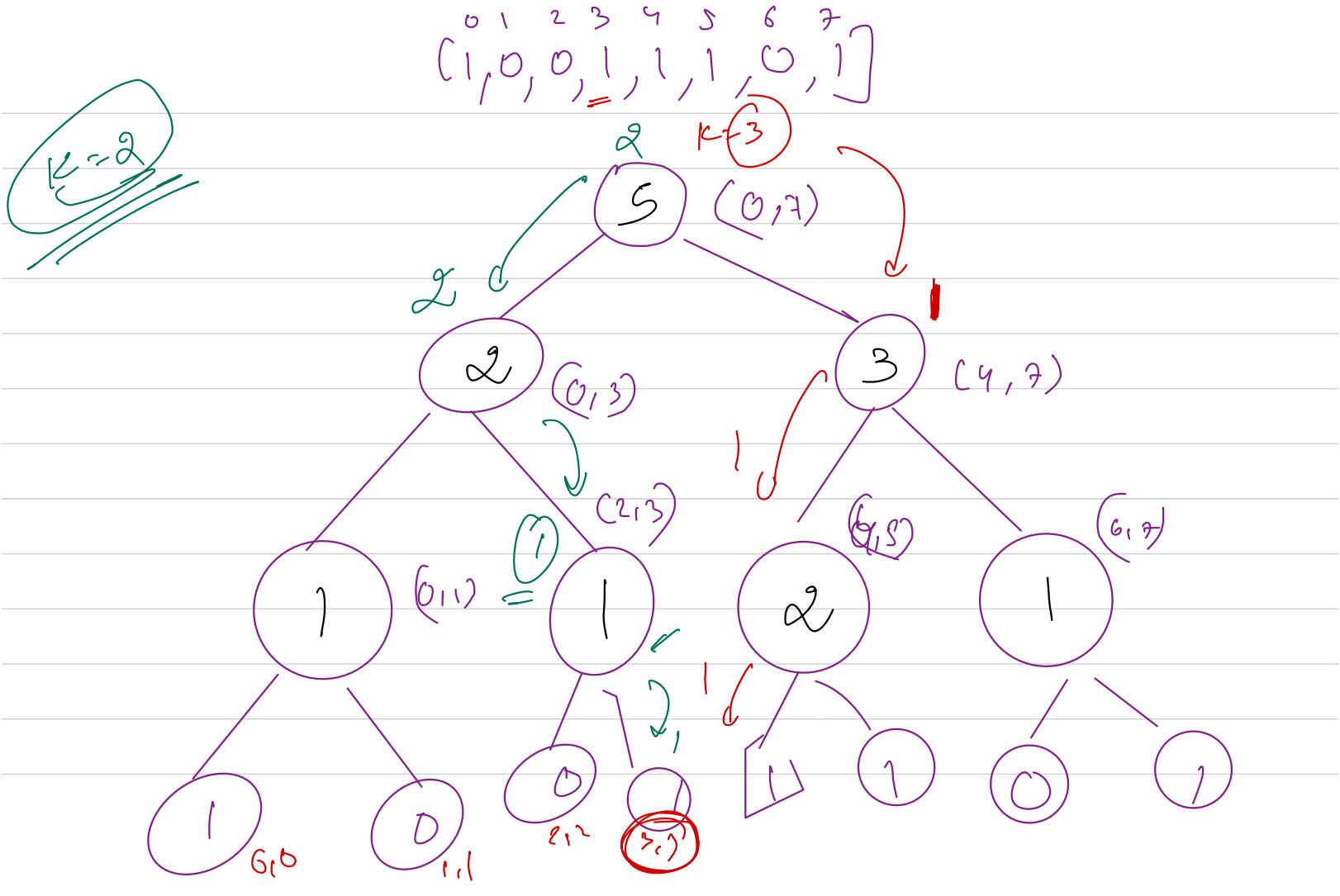
{0,1}

Q) Given an array of binary values. You need to support two different type of queries.

(1) set (i, v) → arr[i] = v $v \in \{0,1\}$

(2) find (k) → return the index of k^{th} one.

find (3)

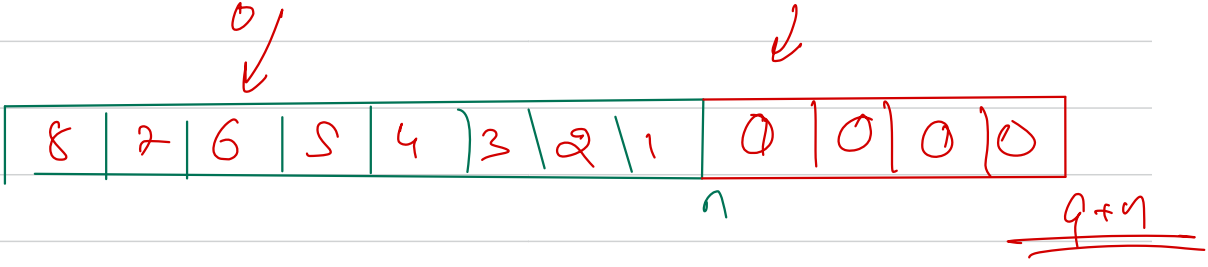

0 1 2 3 4 5 6 7
[1, 0, 0, 1, 1, 1, 0, 1]
K=2
K=3
2
5
(0,7)
2
2
3
(0,3)
(4,7)
1
(2,3)
1
(0,1)
1
1
2
(4,5)
1
(6,7)
1
1
1
0
1
0
1
1
0
1
(0,0)
(1,1)
(2,2)
(3,3)

Q: Jenga Codechef

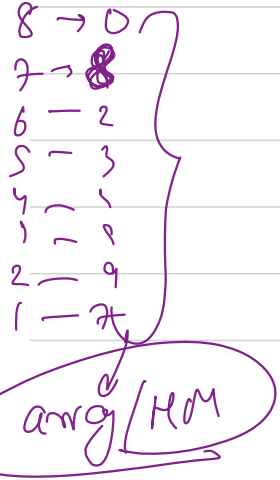
Q: Can we add a new element in a Seg tree ??

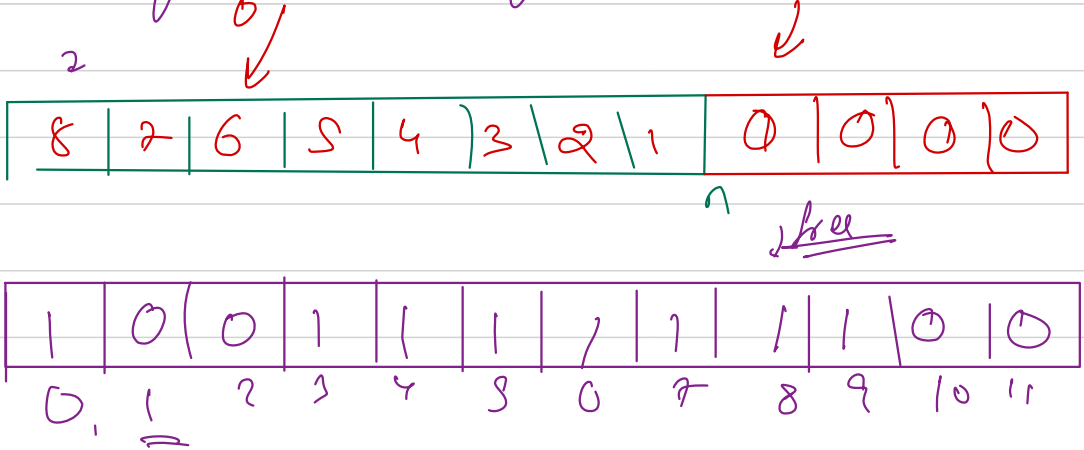
NO

Q: Can we somehow mimic the addition in Seg tree ??



Instead of maintaining the segment of original element at leaf, we will maintain binary value at leaf, and every segment denotes the sum of value of child segment.





Q → Given an array of n integers, and k-queries.

In each query you get 3 values (i, j, k)

for each query with (i, j, k) return the no. of elements greater than k in the subsequence $a_i, a_{i+1} \dots a_j$

$n \le 3 \times 10^4$

$a_i \le 10^9$

no. of query $\le 2 \times 10^5$

n = 6

8 9 3 5 1 9

5

(a, b, c)

i = a xor last_ans

j = b xor last_ans

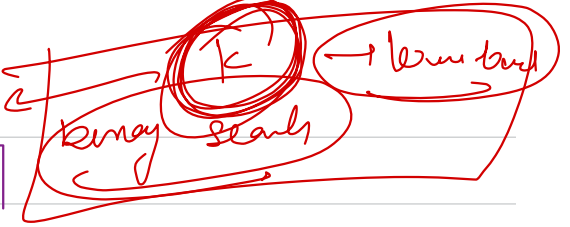
k = c xor last_ans

→ online query

last_ans = 0 → 2 3 5 → 1

→ 3 3 2 → (3^1, 3^1, 7^1)

(i, j) ← segment

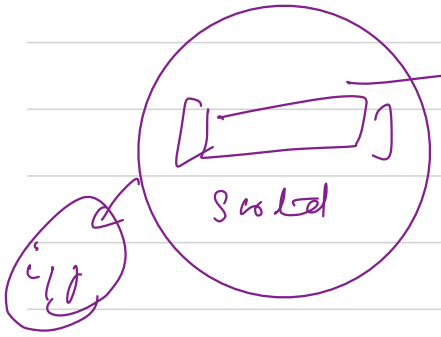


a_1	a_2	a_3	a_4	a_5	a_6	a_7

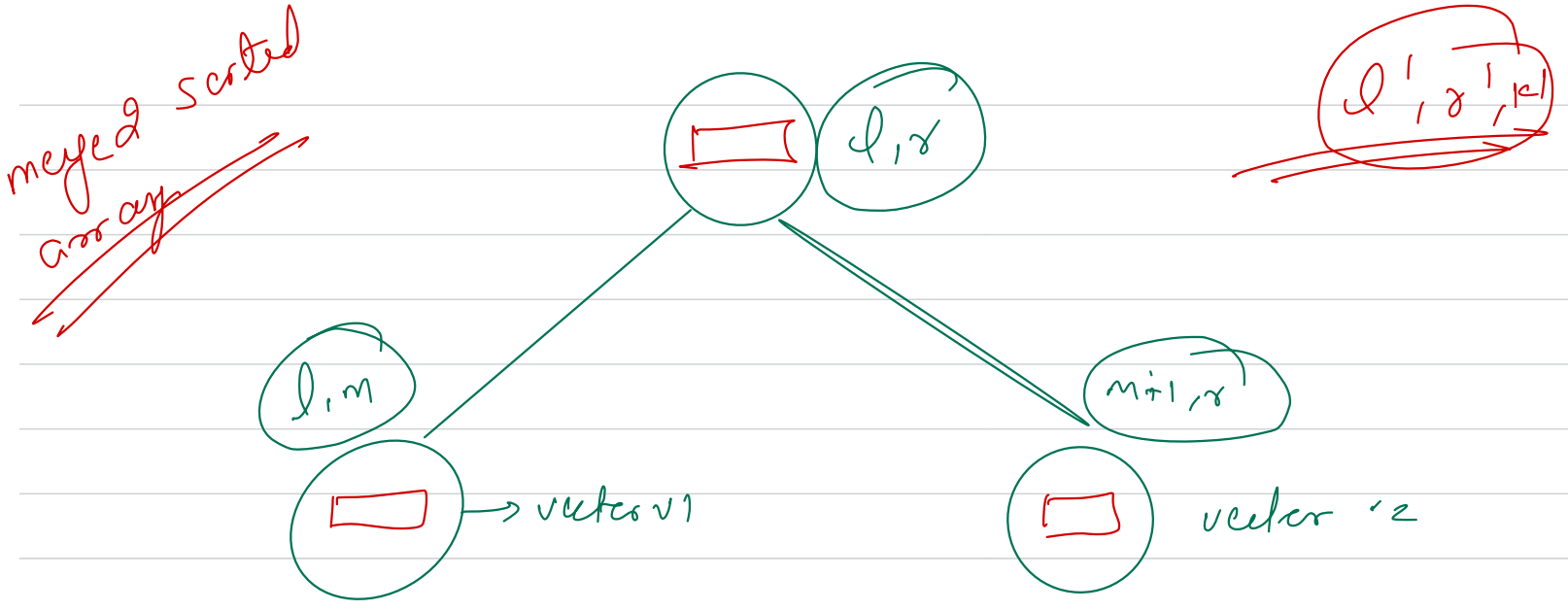
$a_1 < a_2 < a_3 \quad \cdots \cdots < a_n$

For each query in (i, y), we need to look for a Subsequence, and as we only need count, we can modify it.

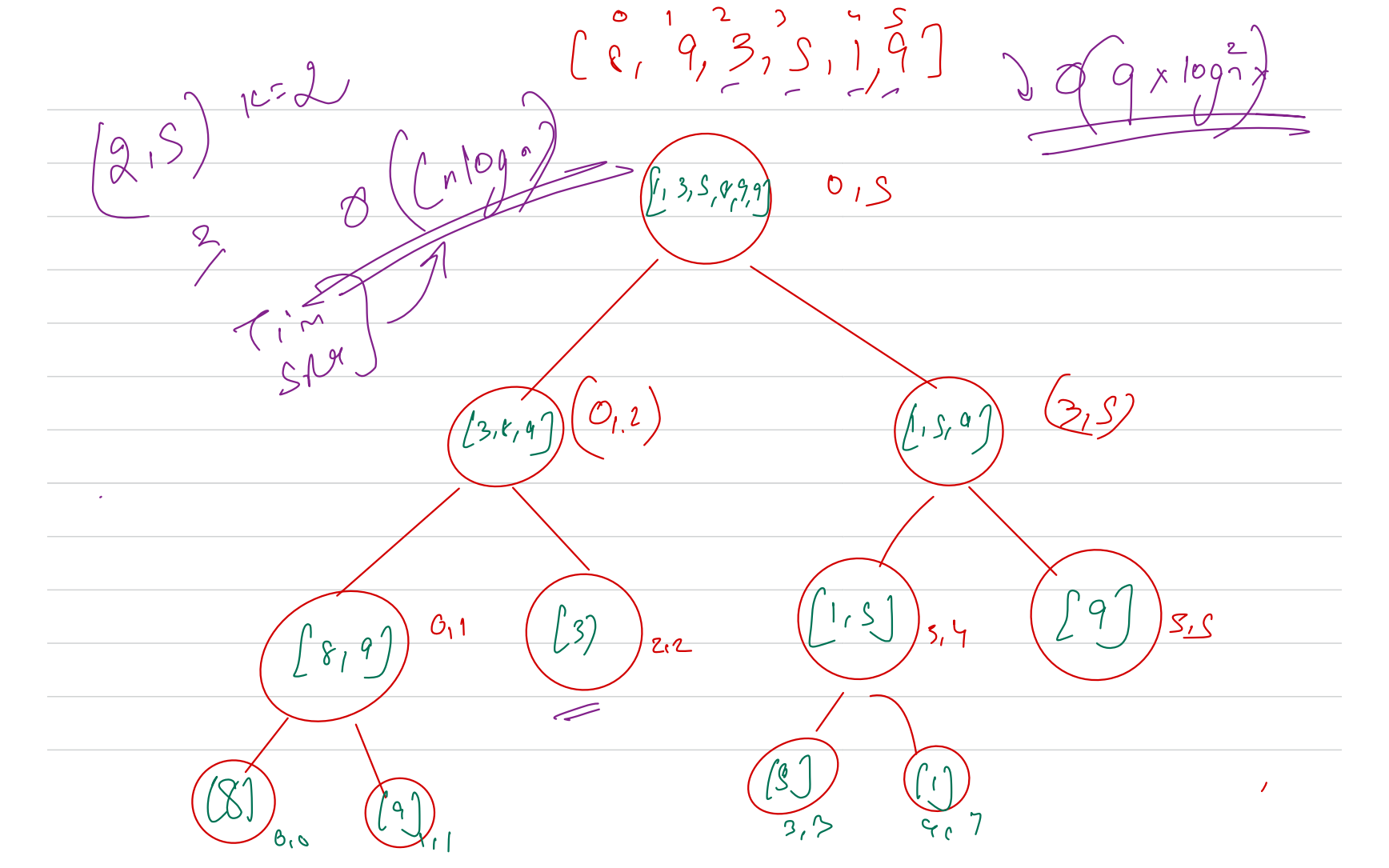
↳ we can use Segment trees.



→ in the node, we will store a sorted vector having elements of the segment


merged sorted array
l, r
l', r', k-1
l, m
m+1, r
vector v1
vector v2

To answer any query, we can just go to calc, the no. of elements greater than k from LST & RST & add them.


[8, 9, 3, 5, 1, 9]
O(9 x log^2 n)
(2,5) k=2
2
O(n log n)
Tim Sort
[1,3,5,8,9,9]
0,5
[3,8,9]
(0,2)
[1,5,9]
(3,5)
[8,9]
0,1
[3]
2,2
[1,5]
3,4
[9]
5,5
[8]
0,0
[9]
1,1
[5]
3,3
[1]
4,4

→ array

vector<ll> tree[4*[ ]]

Q => Quadruples Codechef

0 — 9

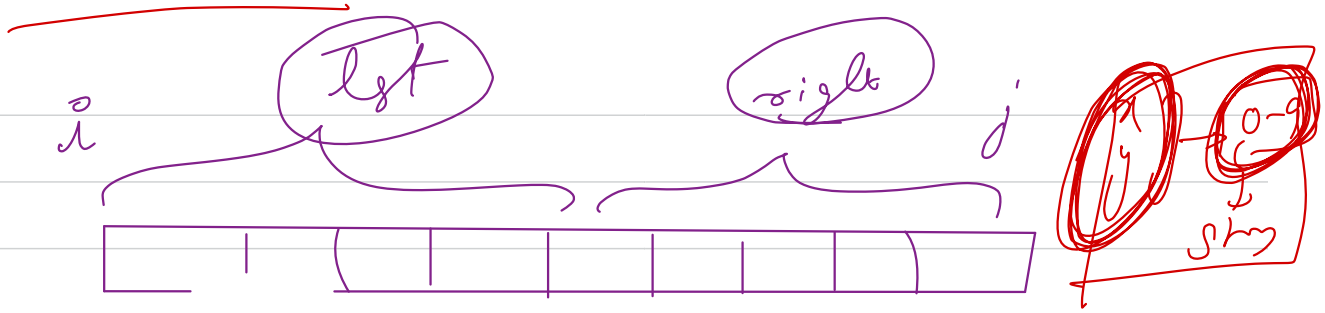
digit

String

(i, j) range

→ (x y y x)

1. (x y y x) + ""
2. "" + (x y y x)
3. xy + yx
4. xyy + x
5. x + yyx



① (xyyx) + ""

② "" + (xyyx)

③ xy + yx

④ xyy + x

③ x + yyx

node LST & RST → # xyyx

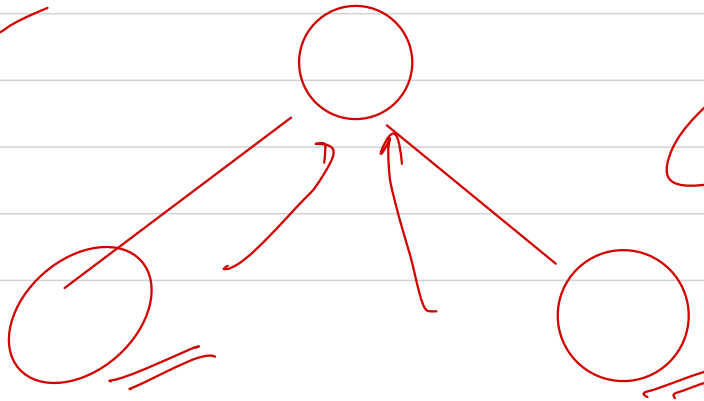
xy

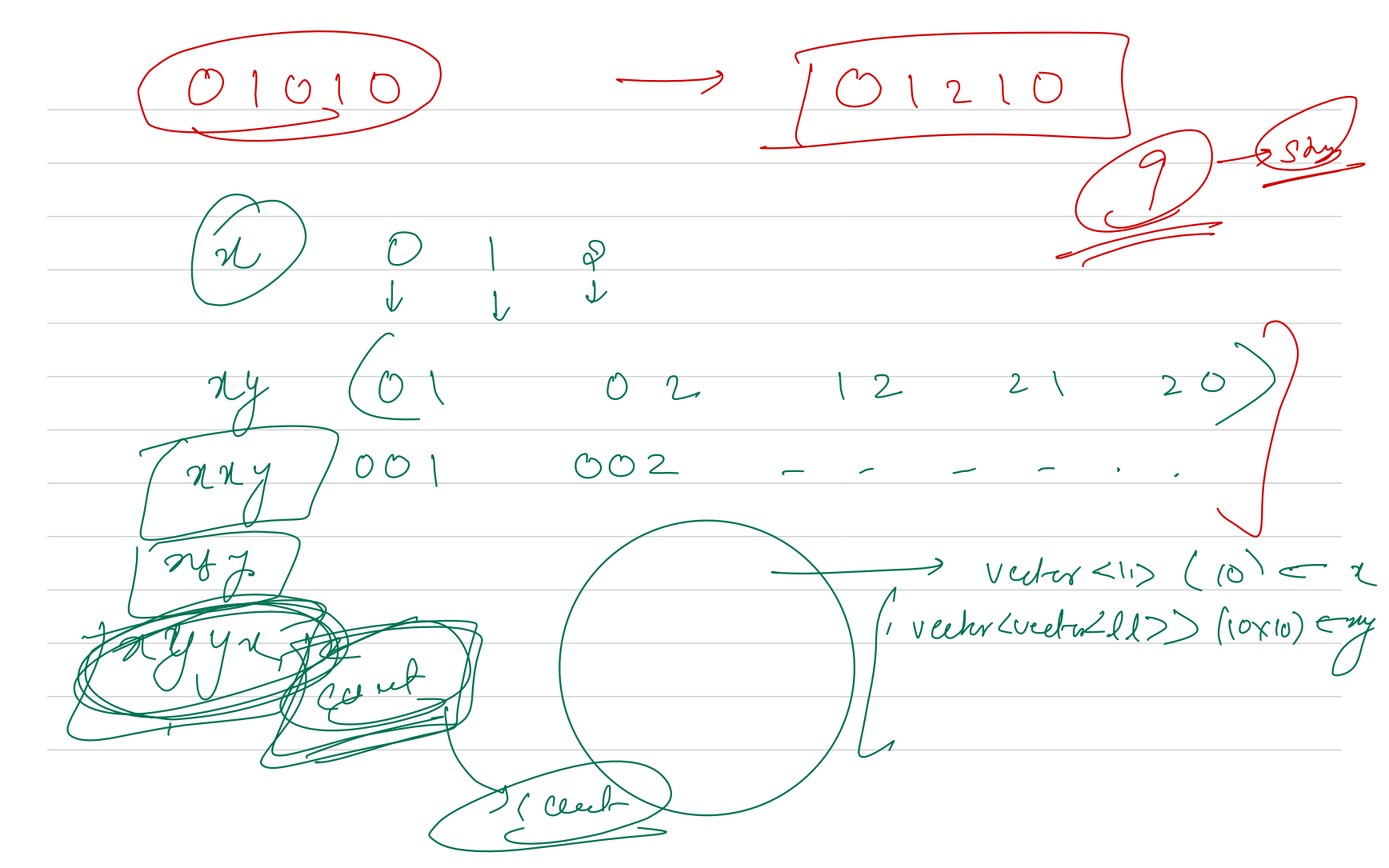
xyy

xxy

x

for(0-9)
for(0-9)




01010
01210
x
0 1 2
xy
01 02 12 21 20
xxy
001 002
xyz
xyyx
count
count
vector<int> (10) <- x
vector<vector<int>> (10x10) <- xy