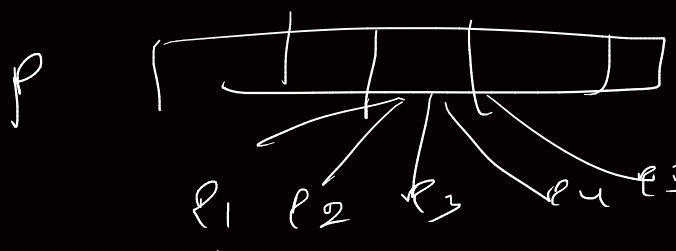
Merge

1) Divide & conq
2) Merging two sorted Arrays..
3) Recursion

Divc



1) Problem P
2) Divide P into small Problem P1, P2, ... Pn

P1 P2 P3 → recursion

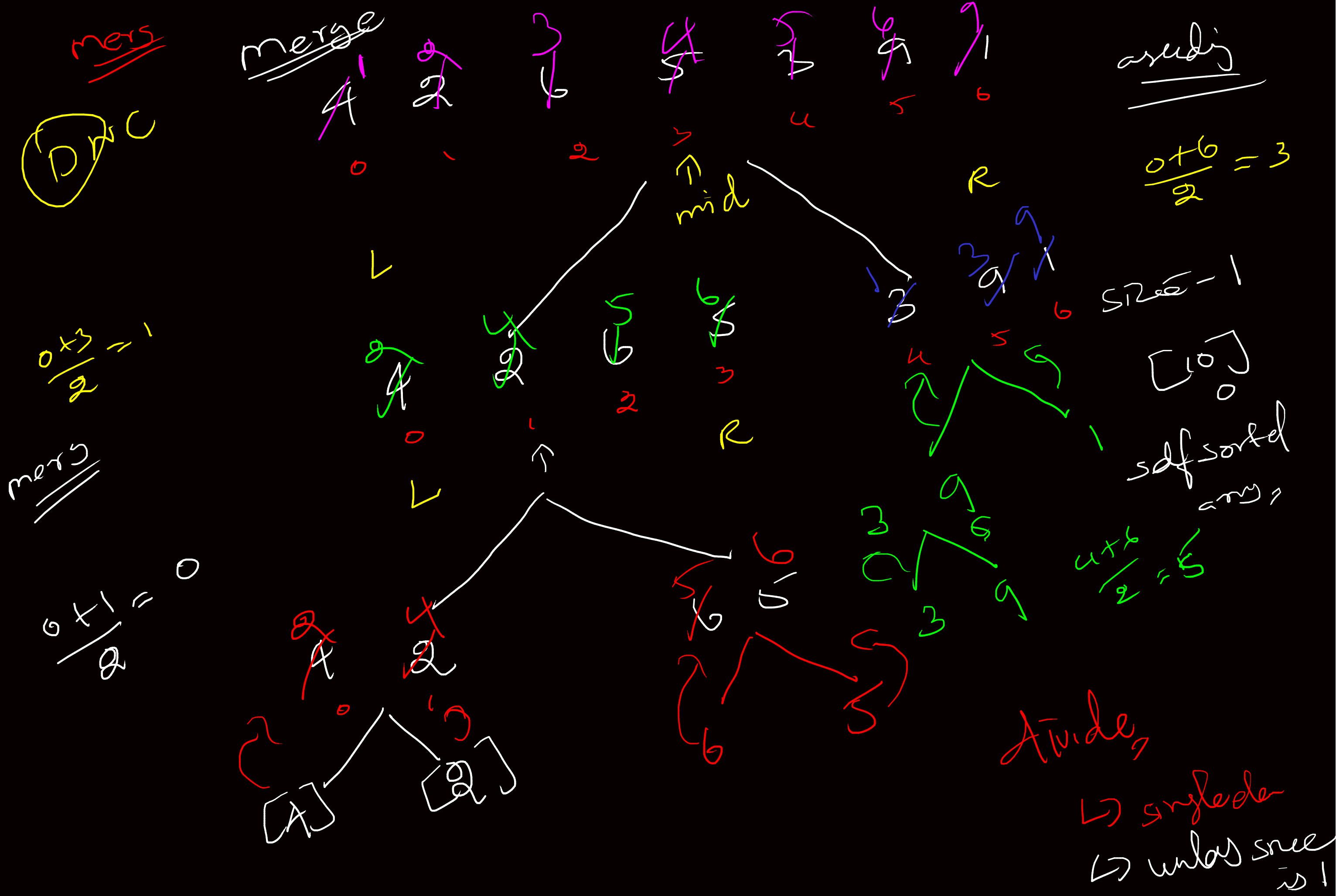
Small P ⇒ Ans

3) Solve every sub indivdly
4) Merge/add all solution to obtain solution for problem P

5! ⇒ 4! * 5

1) Small prob
2) Base case (1! = 1)

very Sml ⇒ !=



dwd merge

arr1 [1 3 5 7 8 13] arr2 [2 4 6 11]

result

[1 2 3 4 5 6 7 8 11 13]

1 2 3 4 5 6 7 8 11 13

[2 8 10 14 16] [3 4 9 11]

2 3 4 8 9 10 11 14 16

Quick sort

1) DNC
2) partition

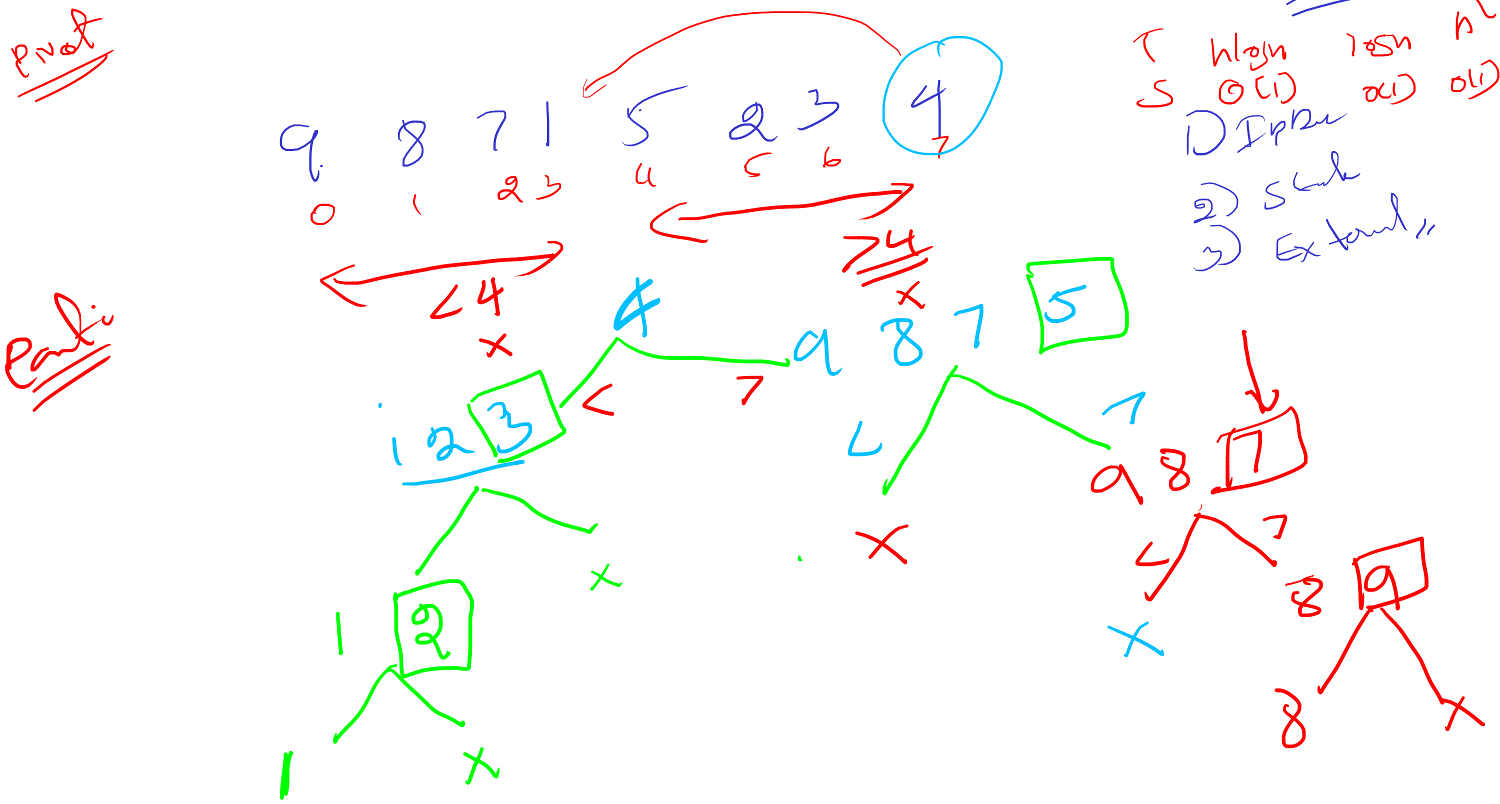
merge sort

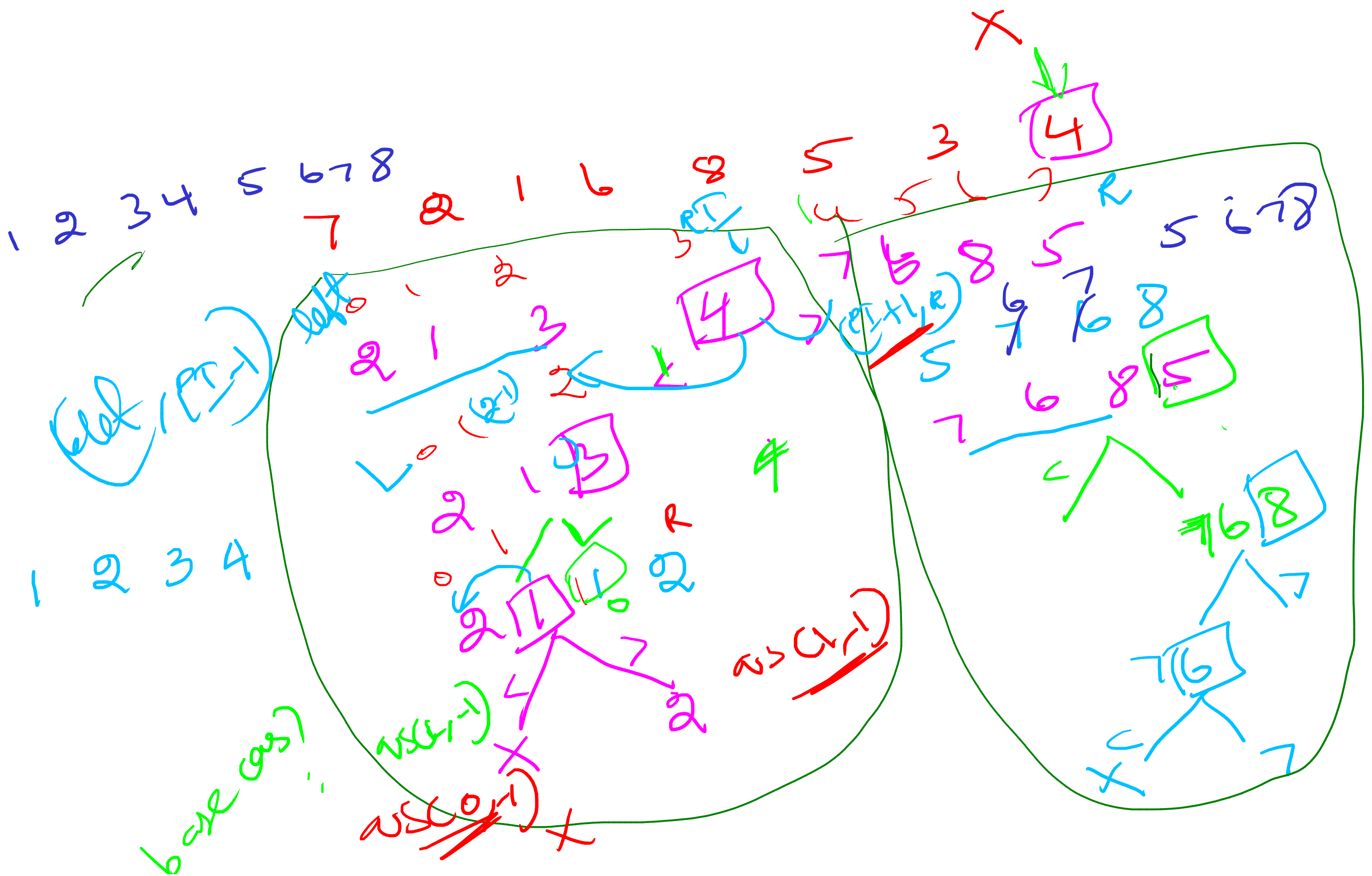
	B	A	W
T	O(nlogn)	nlogn	nlogn
S	O(n)	O(n)	n

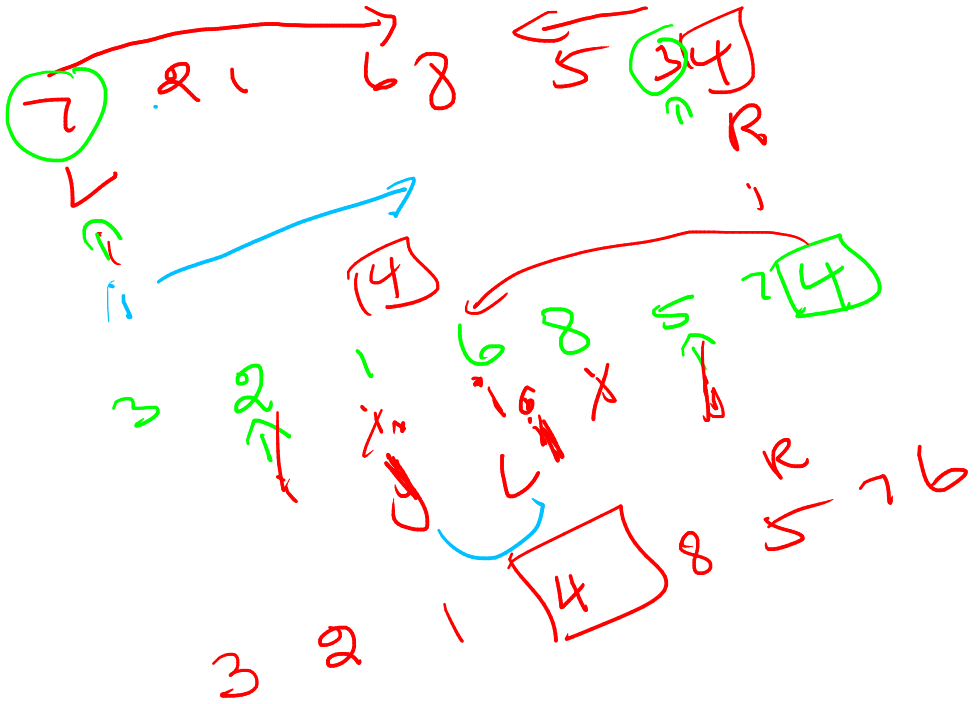
Quick sort

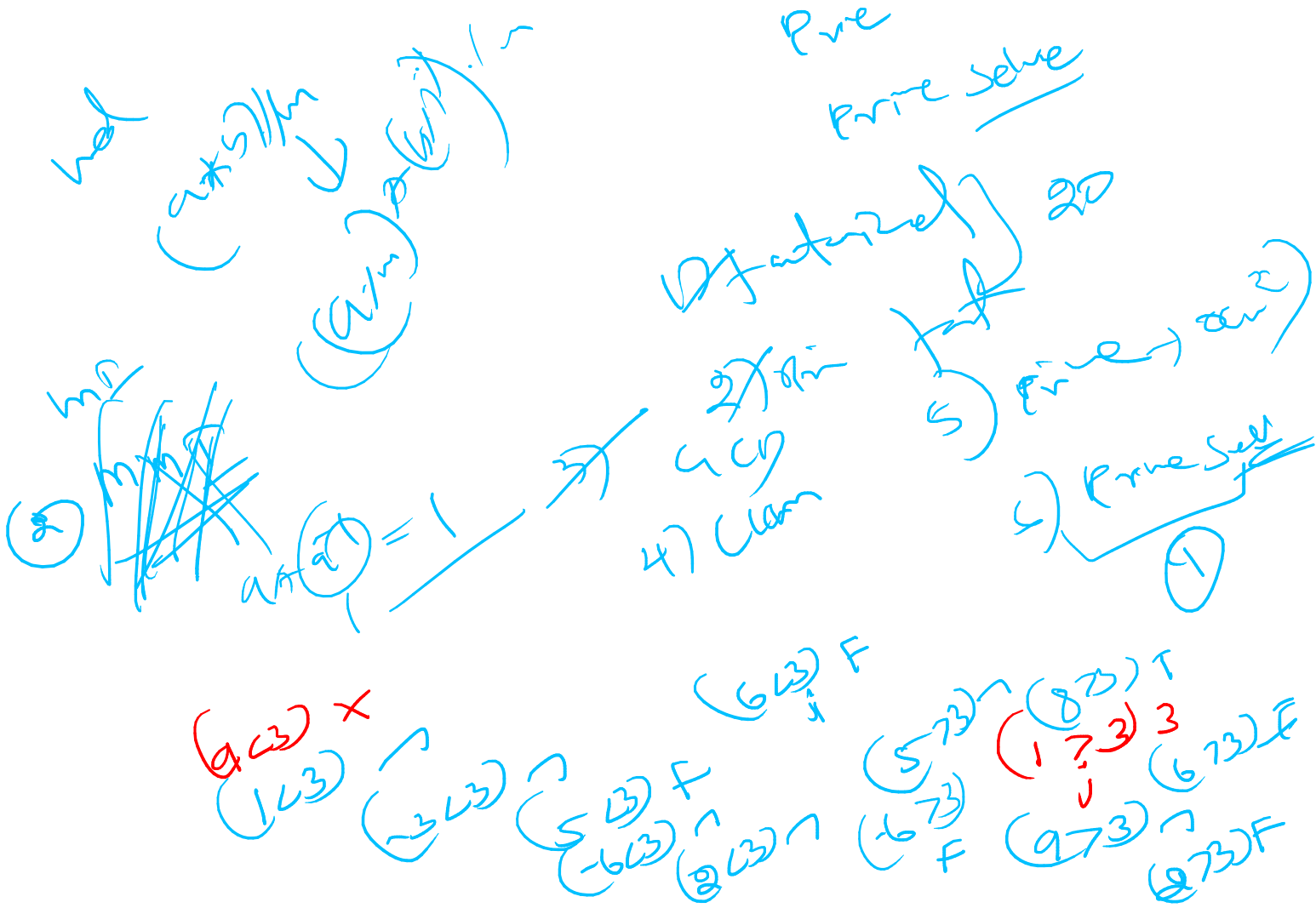
T	nlogn	logn	n^2
S	O(1)	O(1)	O(1)

1) Inplace
2) Stable
3) Extend



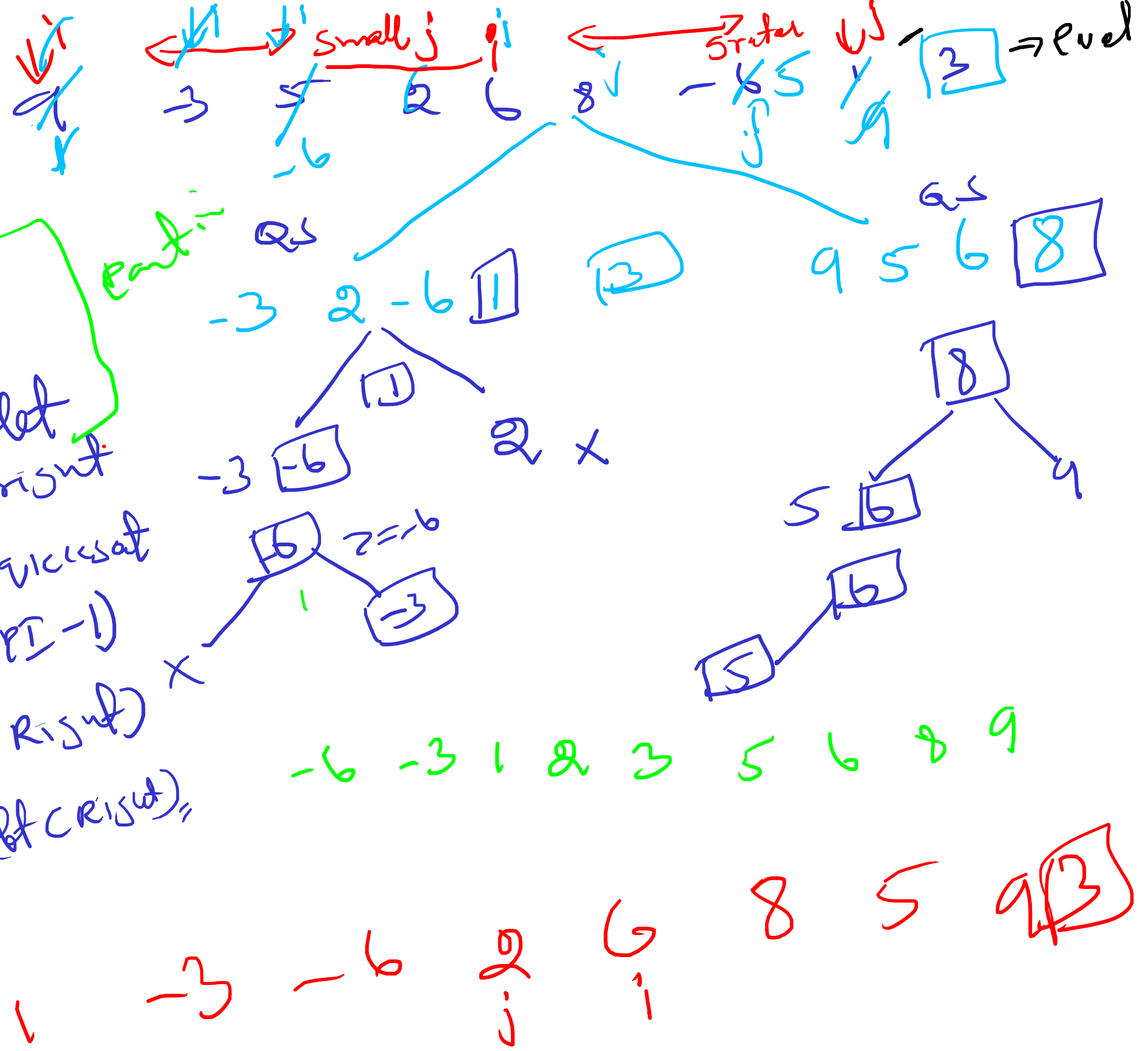


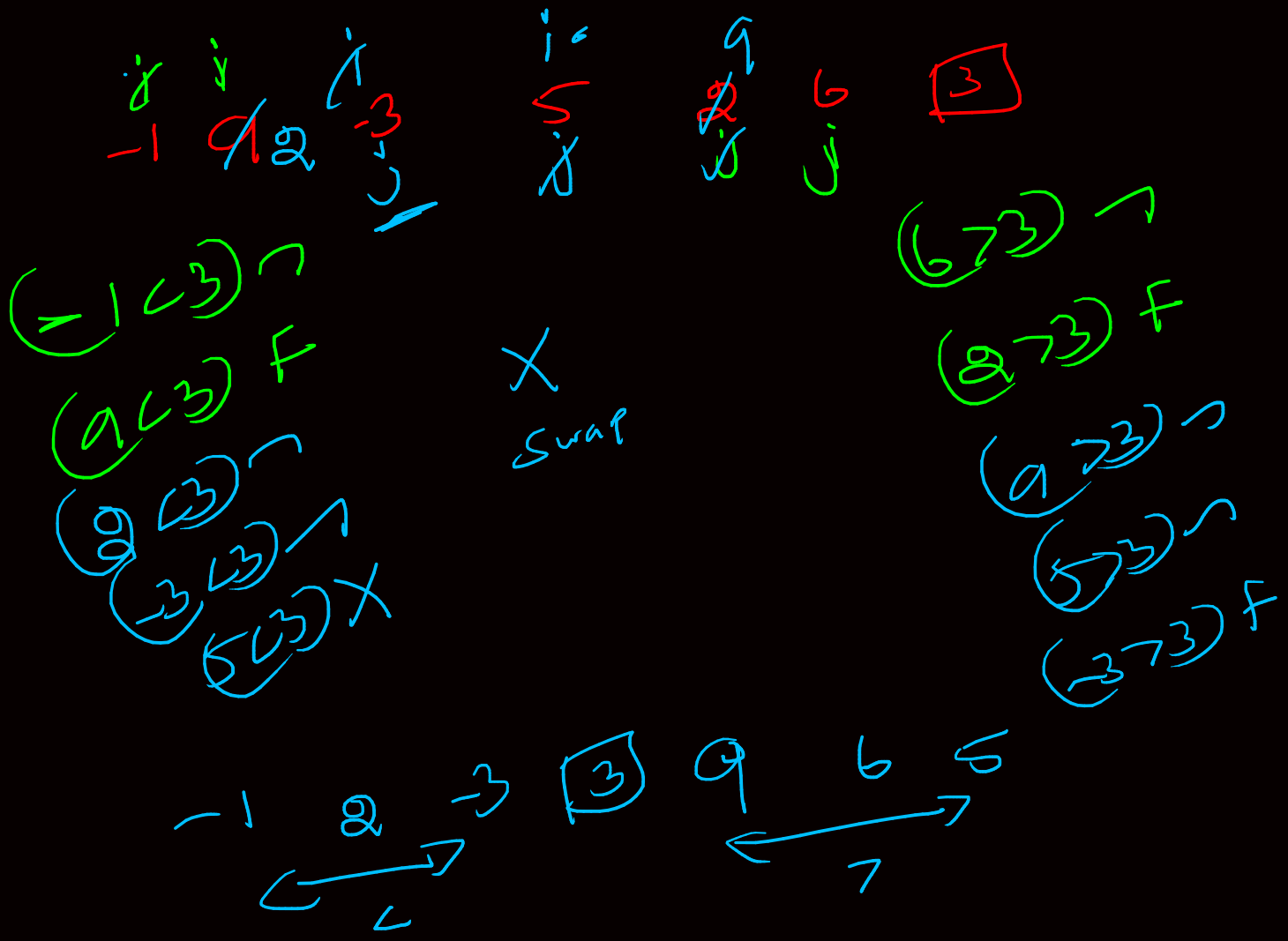




QS

1) find pivot
2) reorder arr
   < piv → left
   > pivot → right
3) again do quicksort
   QS (left, PI − 1)
   QS (PI+1, Right)
   until (left < Right)





1) Fault
2) True fault
3) is prime
4) True Sieve

Factors

10

2 x 5 = 10

1 x 10 = 10

2 5

1 10

Divisor

10

1 2 5 10

1 to 10

if (n % i == 0)

i

Q)

n = 23 45 678 910

10 % 1 == 0

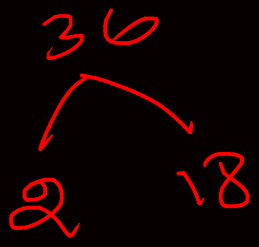
10 % 3 = 1

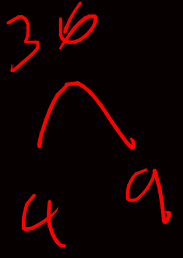
10 % 4 = 2

5 % 4 = 1

Remainder

36





3 * 12

36

1 to $\sqrt{36}$

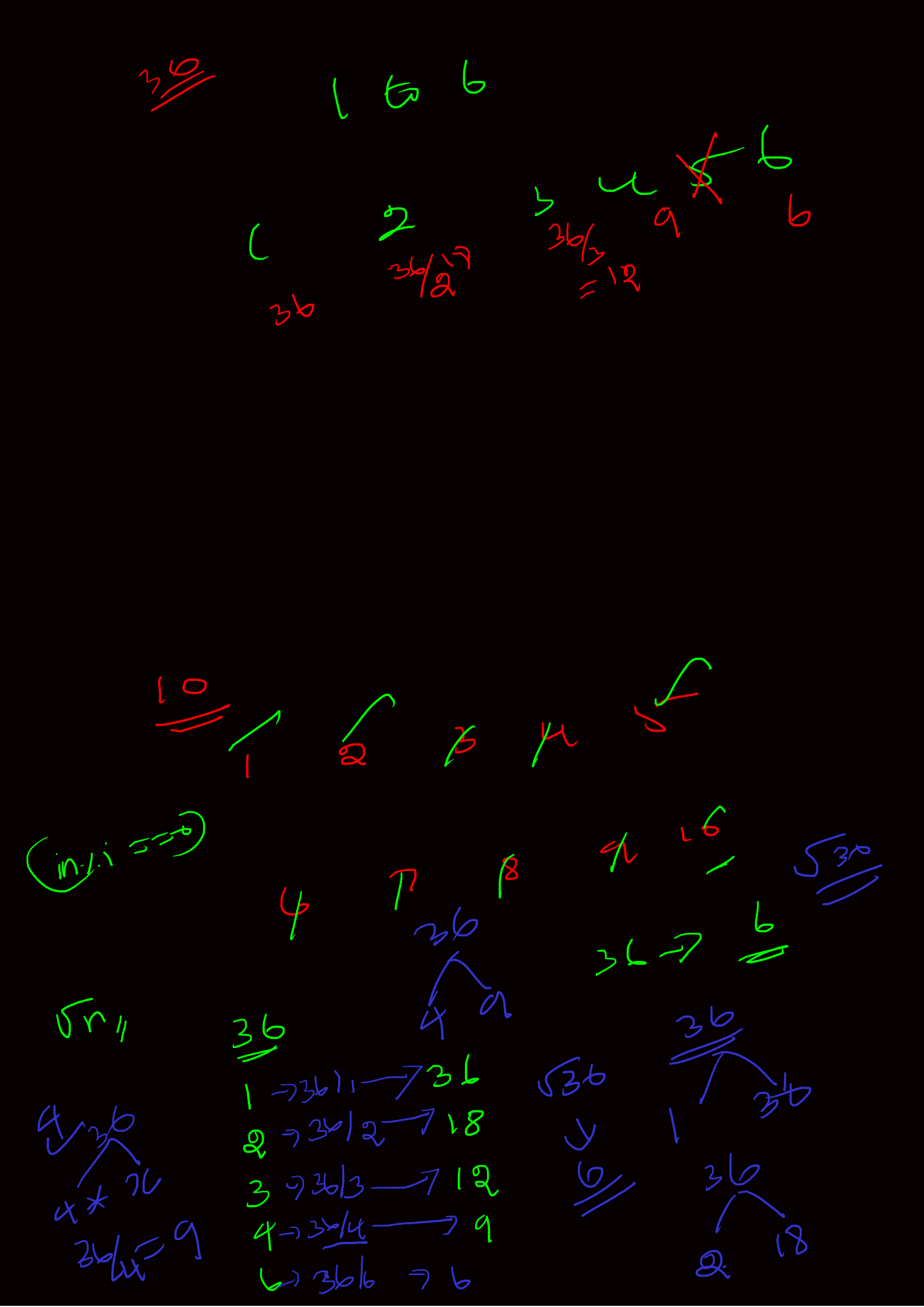
1	36
2	18
3	12
4	9
6	6

Sort(3)

36/2 = 18

36/3 = 12

36/4 = 9



1) factors
2) prime factors
3) isprime
4) prime seive.

modulo :-

$7 \% 5 = 3$

rem = 2

$18 \% 5 = 3$

$19 \% 5 = 4$

$24 \% 3 = 0$

rem

$x \% 5$

$(0, 1, 2, 3, 4)$

$x \% n \hookrightarrow [0, n-1]$

$(a+b) \% c$

$(10+20) \% 4$

a 10
b 20
c 4

$30 \% 4 = 2$

$(a+b) \% c = ((a \% c) + (b \% c)) \% c$

$= ((10 \% 4) + (20 \% 4)) \% 4$

$= (2 + 0) \% 4$

$= 2 \% 4 = 2$

int => (10^9)

10^9

(10^9+1) min

range +1

$(10^9 + 5)$ %4

a b

-10^9-4

min max

int

min

-10 -9 -8 -7 . . . . 0 1 . . 1 . . . 2 . . 10 11

% 10^9 +7

a = 10

a = a+1 (11)

$(10+2)$ %4 => ((10%4) + 2%4) %4

$(2+2)$ %4 = 0

a = 10 %4

b = 2

5! 120 ans=1 1

(1*1) % md

(1*2) % md

(2*3) % md

(6*4) % md

(24*5) % md

(120*6) % md

120

1=2 3 4 5 6

a^9 => => -ve

100!

$a^9!$ % x

x

47

Q 720 % x

Q

$(a+b)\%c = ((a\%c) + (b\%c))\%c$

$(a*b)\%c = ((a\%c) * (b\%c))\%c$

$(a-b)\%c = ((a\%c) - (b\%c))\%c$

$a = 13$
$b = 4$
$c = 5$

$(13-4)\%5$

$9\%5 = 4$

$((13\%5) - (4\%5))\%5$

$(3 - 4)\%5$

$(-1\%5)$

$(a-b)\%c = ((a\%c) - (b\%c) + c)\%c$

$((13\%5) - (4\%5) + 5)\%5$

$(3 - 4 + 5)\%5$

$4\%5 = 4$

$((a\%c) - (b\%c) + (c\%c))\%c$

$(a/b)\%c = \left(\frac{a\%c}{b\%c}\right)\%c$ ✗

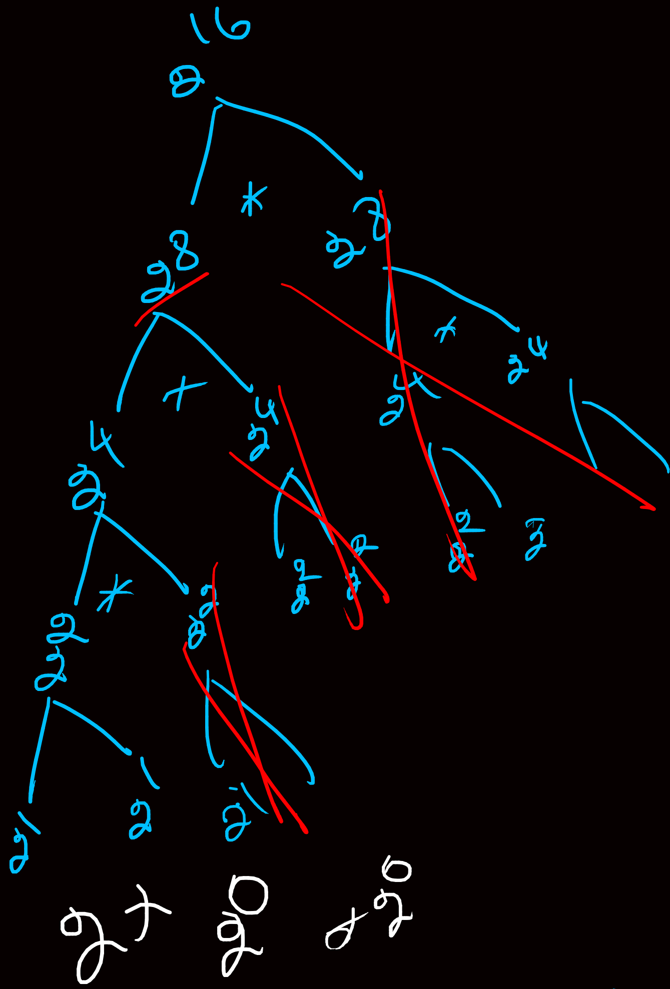
$x\%c \rightarrow (0 \ldots c-1)$

$b\%c = (0 \ldots c-1)$, e/q $a \neq 0$

$(a+b)\%c = ((a\%c) + (b\%c))\%c$

$(a/b)\%c = \left(\frac{a\%c}{b\%c}\right)\%c$

a=2
b=16
b/2

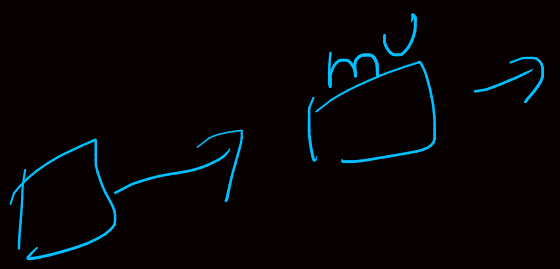


Bit manipulation

Bi nul

(0,1)

OFF
ON



Decimal

$(341)_{10} \Rightarrow 3*10^2 + 4*10^1 + 1*10^0$

$\Rightarrow 300 + 40 + 1$

$= 341$

[0 1 2 3 ... 9]

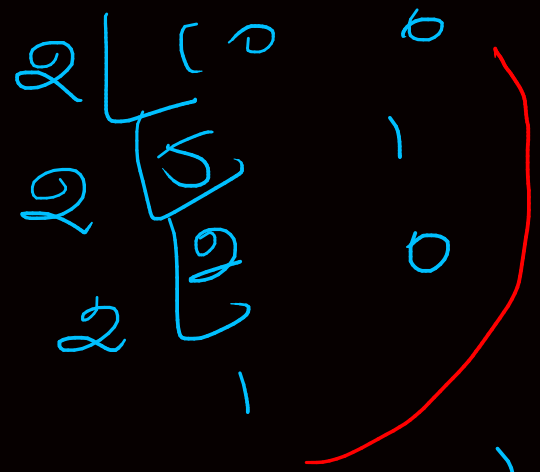
Binary → Decimal

$(101)_2 \Rightarrow 1*2^2 + 0*2^1 + 1*2^0$

$= 4 + 0 + 1$

$= 5$

$(101)_5$



8 4 2 1

2^3 2^2 2^1 2^0

1 0 1 0

Decimal

$8+0+2+0$

1 1

1 6 1

1 1 1

1 1 0 0

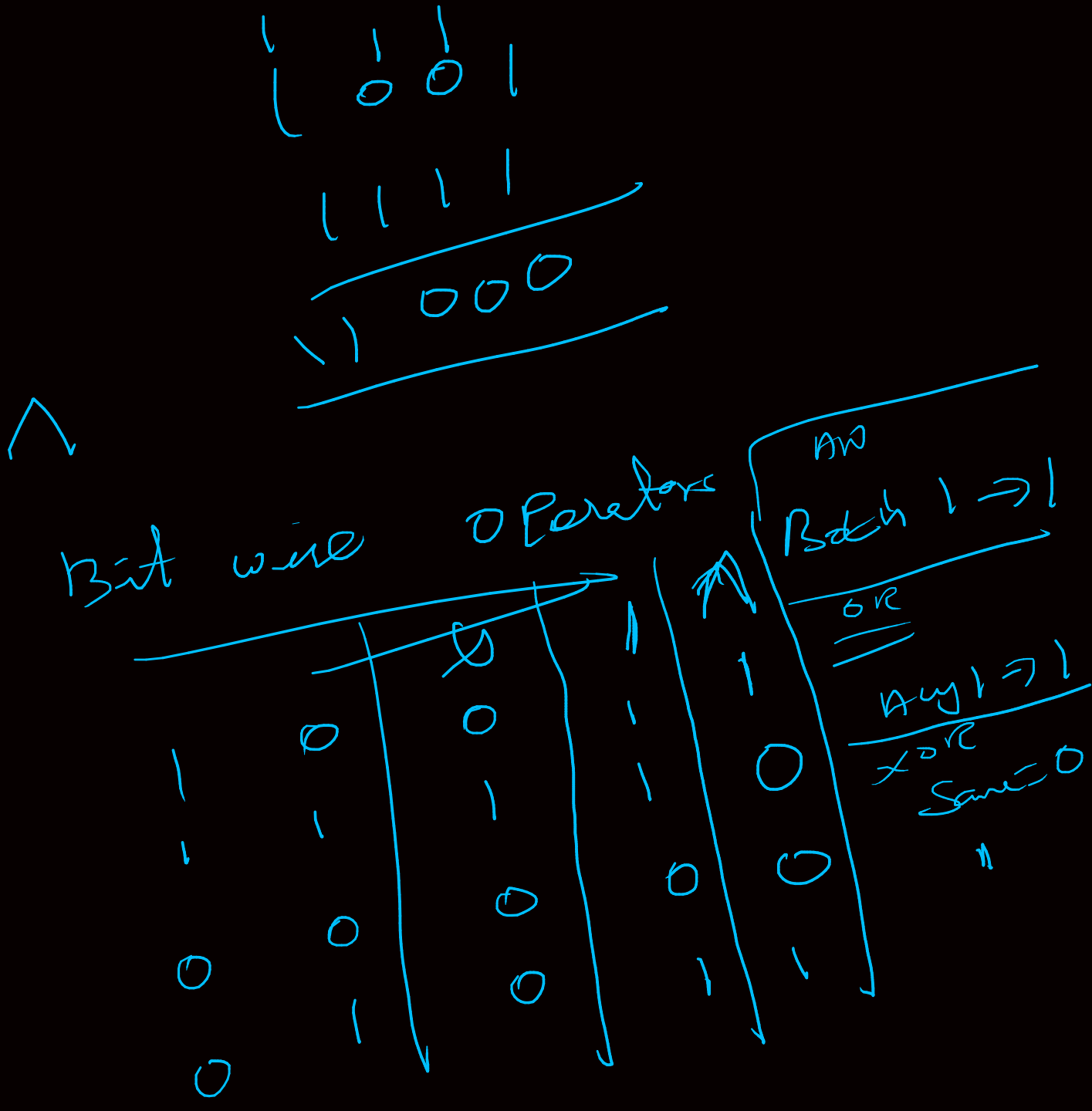
275

$\Rightarrow$ 47

6 2 2

$1+1 \Rightarrow 10$

$1+1+1 = 11$


1 0 0 1
1 1 1 1
1 1 0 0 0
∧
Bit wise Operators
AND
Both 1 ⇒ 1
OR
Any 1 ⇒ 1
XOR
Same = 0
1 0 0 1
1 0 1 0
0 0 0 0
0 1 0 1
1 ∧ 1 = 0
Same = 0
diff = 1

n

n : 7

[1 5 2 3] ④

so ins missing

n = 6 (1 to 6)

(n-1) sub [1 2 4 5 6]

③

missing

n = 4 @ 4

[1 2 4] = ③

O(n)

(n log n)

Bit Base will be fast

a b
10 - 6 = 0

h = 5
arr = [1, 2, 4, 5]

10 ^ 10 = 0

x = 0

c(n 2) 3 4 5

10 ^ 10 = 0

sub 6 =

(1 2, 4, 5)

(3)

n = 4 [1 3 4] ②

1) Step 1 to N

a = (1 ^ 2 ^ 3 ^ 4) ⇒

2) take one by one from a or do xor with a

(1 ^ 3 ^ 4)

(1 ^ 2 ^ 3 ^ 4) ^ (1 ^ 3 ^ 4) = ②

Same = 0 diff = 1

~ 1 → 0

0 → 1

Not (101) ⇒

010

011 → 100

Right shift   Left shift.

>>   <<

101 >> 1   10
010   10

Right shift

101 → 010

101 → 10

32

65 4 3 2 1 0
101 → 000010
00 0 0 0

Right

12

8 4 2 1
1100 → 12

8 4 2 1
1) 0110 → 6

8 4 2 1
2) 0011 → 3

16 >> 1 = 8

16 & 1 = 0

10000 & 00001 = 00000

2 => 010

2 << 1 => 100 = 4

4 << 1 = 1000 = 8

n >> 1

n << 1

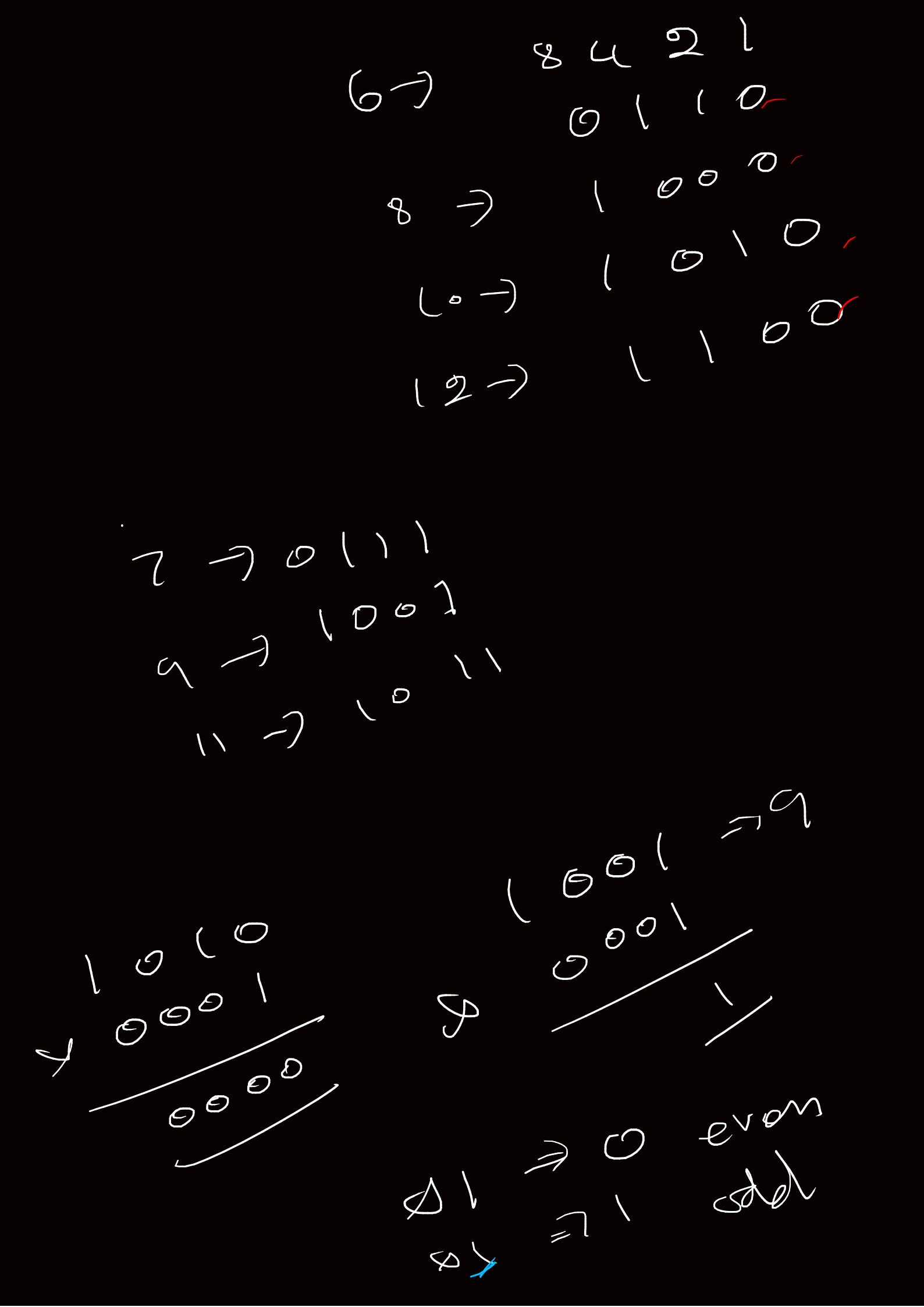
n & 1

odd / even

if (n & 1) == 0

3 & 1 = 1

odd


8 4 2 1
6 → 0 1 1 0
8 → 1 0 0 0
10 → 1 0 1 0
12 → 1 1 0 0
7 → 0 1 1 1
9 → 1 0 0 1
11 → 1 0 1 1
1 0 1 0
- 0 0 0 1
0 0 0 0
9 → 1 0 0 1
0 0 0 1
1
01 → 0 even
11 → 1 odd

1077 !

Swap 2 vari but :

Swa a/b

a = a^b 2
b = a^b 5
a = a^b 7

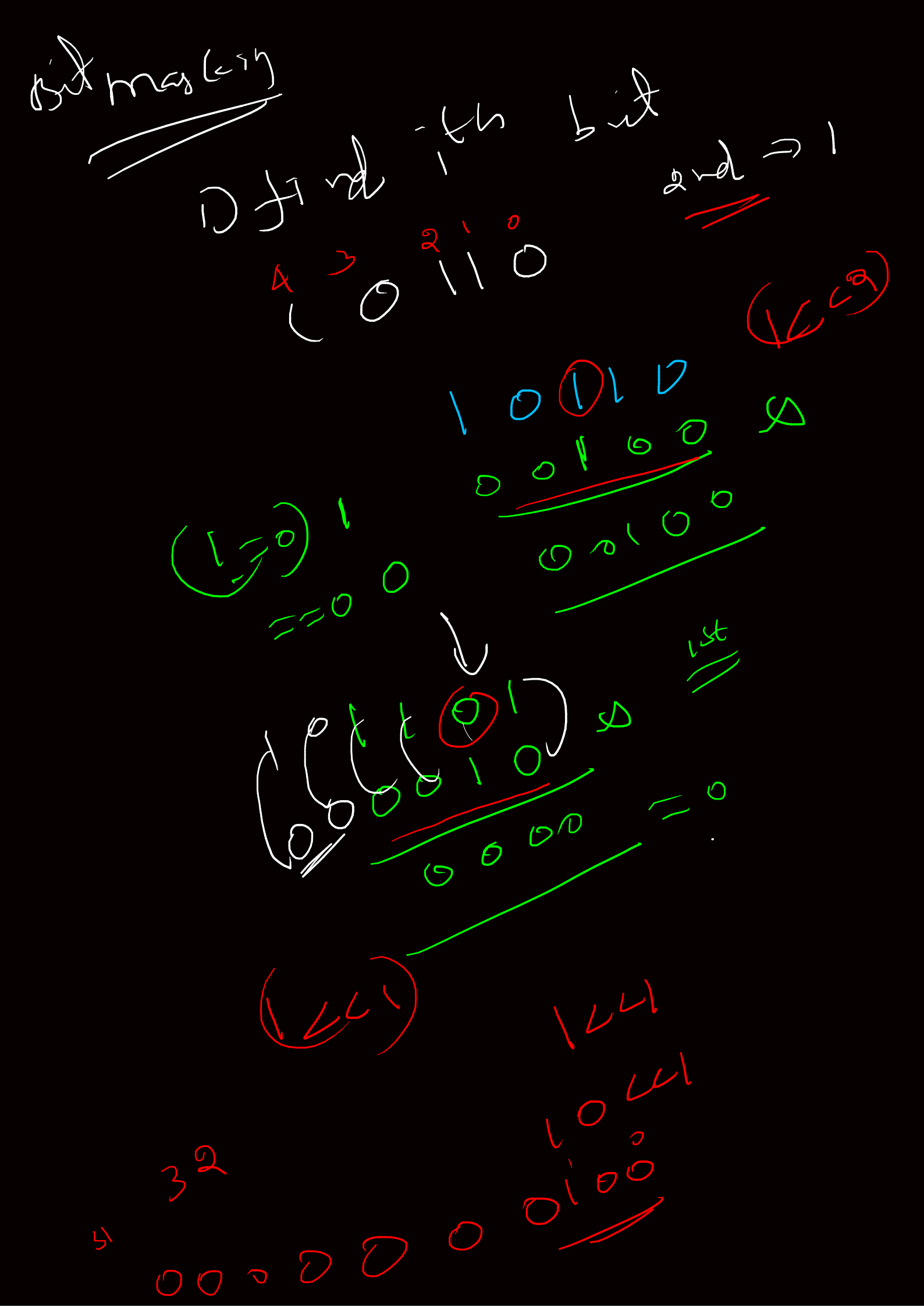
a = 5 2
b = 1 5

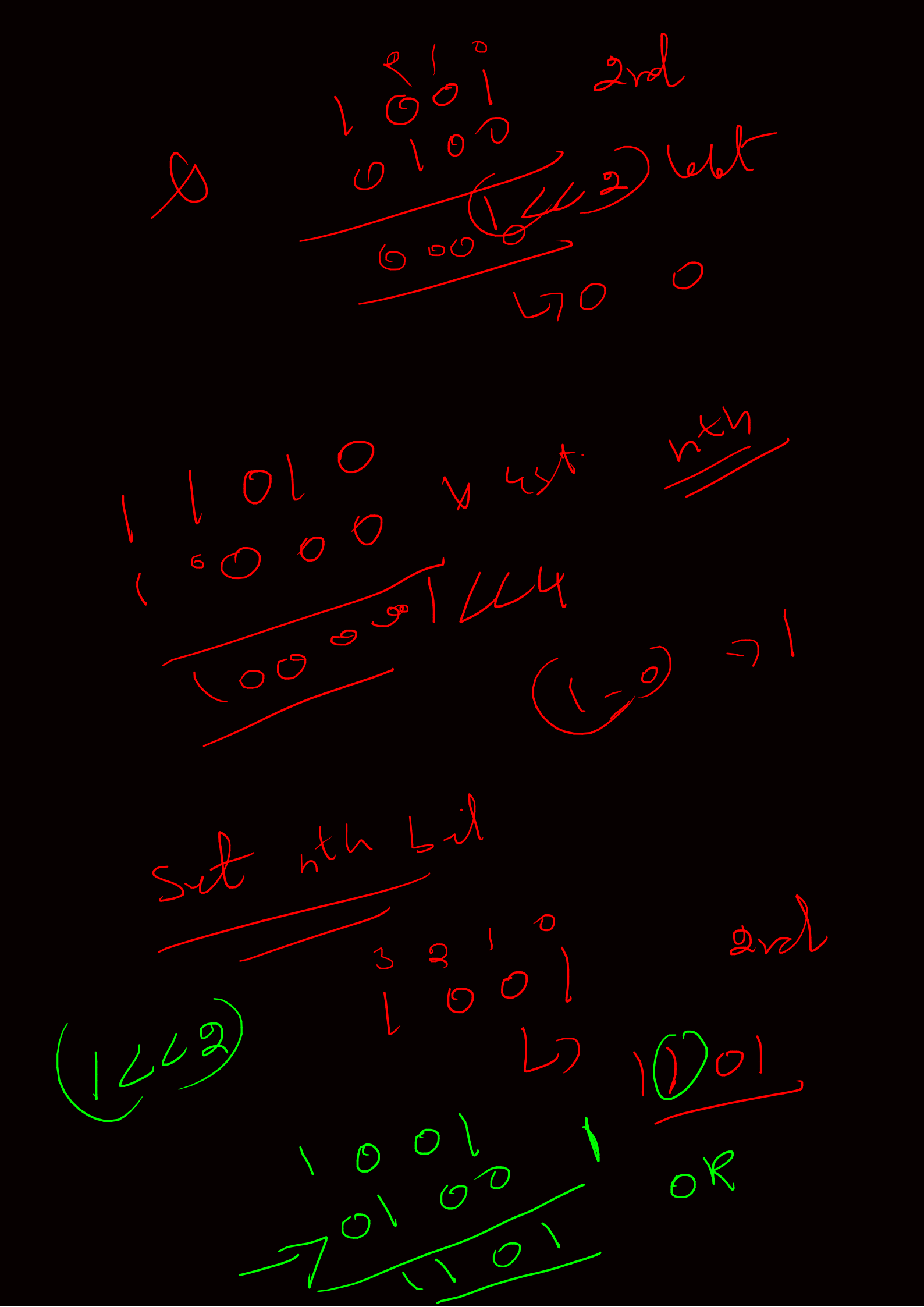
a =

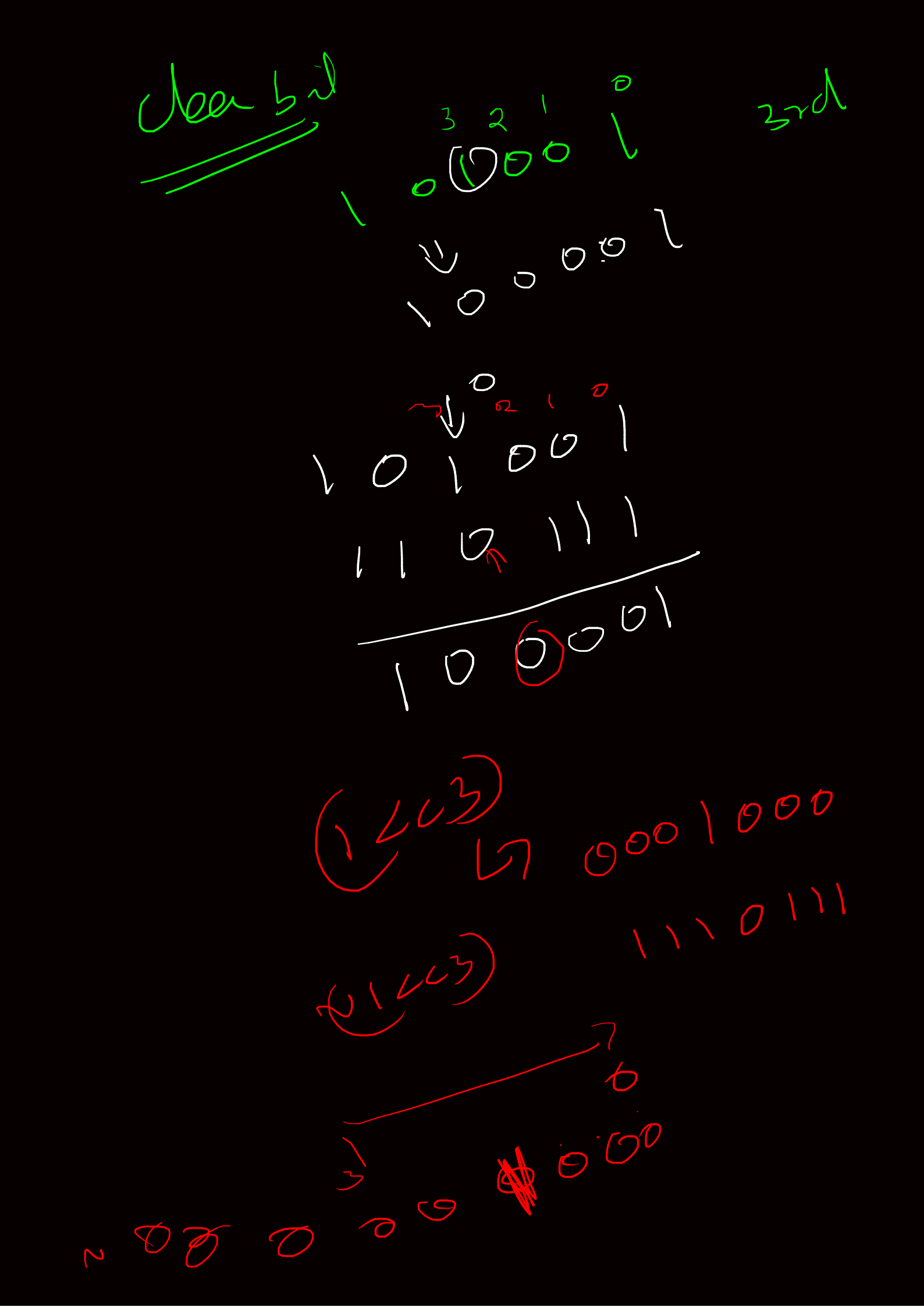
421
a = 101
b = 111
010

010
101
111

010
111
101 → 5


bit mask
Find i-th bit
and = 1
(1<<3)
1 0 1 1 0
0 0 1 0 0
0 0 1 0 0
(1==0)
==0
(1<<2)
1st
=!0
(1<<1)
1<<1
1<<1
0 0 0 0 0 1 0 0


2nd bit
nth bit
Set nth bit
2nd


Clear bit
3rd
1 0 0 0 1
1 0 0 0 1
1 0 1 0 0 1
1 1 0 1 1 1
1 0 0 0 0 1
(1<<3)
0 0 0 1 0 0 0
1 1 1 0 1 1 1
~(1<<3)

Bit masking both "1" -> 1 "0".

1) Find ith bit

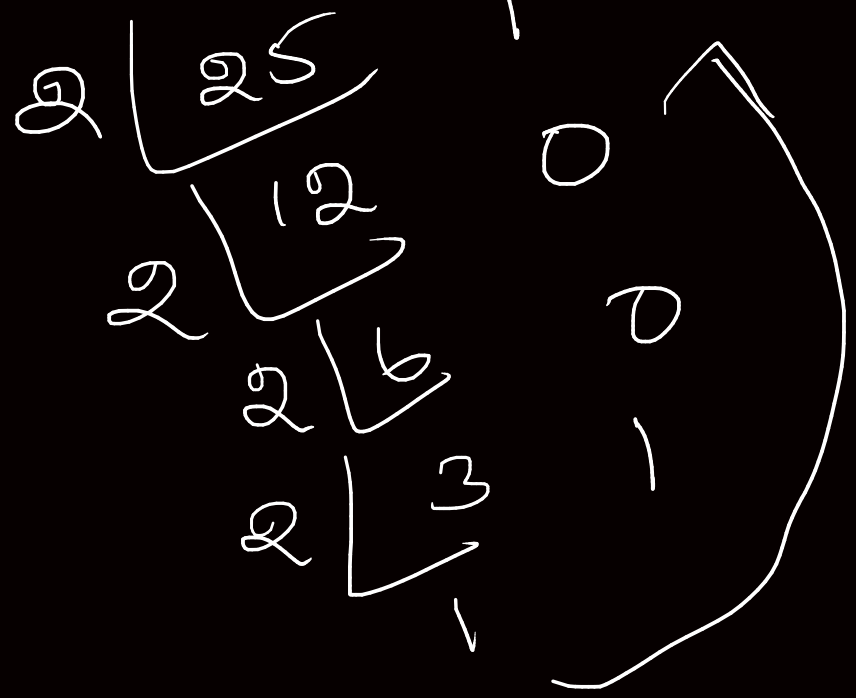
number & (1 << i)

2) set ith bit

number | (1 << i)

3) clear

number & (~(1 << i))



11001

1) Bit mask
2) Prefix Array
3) Longest subarray Sum
   → Kadane's Algo

Bits 0/1

$(23)_{10} = 1\times10^2 + 2\times10^1 + 3\times10^0$

$(01)_2 = 1\times2^2 + 0\times2^1 + 1\times2^0$

$= 4 + 0 + 1$

$= 5$

Convert Decimal → Binary

Divide 2

2	17	1
2	8	0
2	4	0
2	2	0
	1	

$2^4\ 2^3\ 2^2\ 2^1\ 2^0$

16 8 4 2 1

10001

1 1		
0 1		
1 1 1		
1100	10	11

both 1

0 0	0	
0 1	0	
1 0	0	
1 1	1	

OR any one

1
0
1
1

XOR same → 0, diff → 1

0
1
1
0

~ ⇒ 1 → 0, 0 → 1

~101 → 010

?? 7 → 111

Right shift ?? / 2

111 → 011 (3) 7/2 = 3

Left shift * 2

111 → 1110 → 7*2 = 14

10 Right shift ↷ 5

8 4 2 1
1 0 1 0
0 1 0 1

8 4 2 1
0 1 0 1

Left shift ↶ 20

10

8 4 2 1
1 0 1 0

16 8 4 2 1
1 0 1 0 0

Prefix Array

Sum

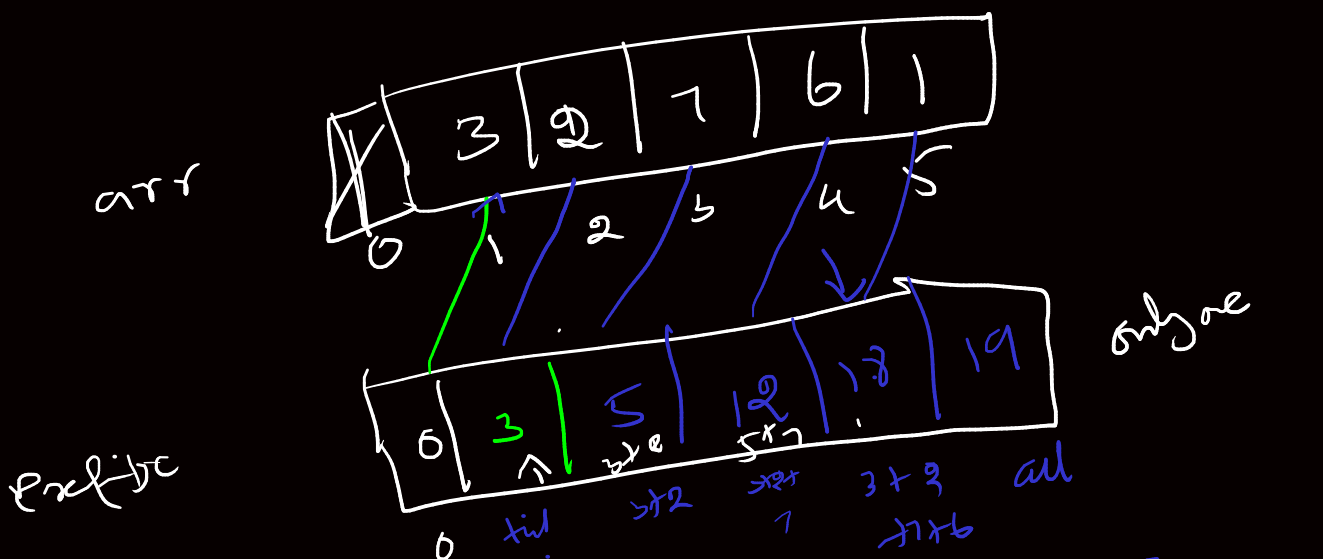
1	7	3	4
1	2	3	4

q → queries
L ≤ R

1 3 → sum[1...3] 1+7+3 = 11

3 4 → sum[3...4] 3+4 = 7

2 4 → sum[2...4] 7+3+4 = 14



L R
2 4

2+7+6 = 15

P[R] − P[L−1]
P[4] − P[2−1]
13 − 3 = 15

P[3] − P[0]
7−0 = 7

L R
1 5

O(1)

max array → on right

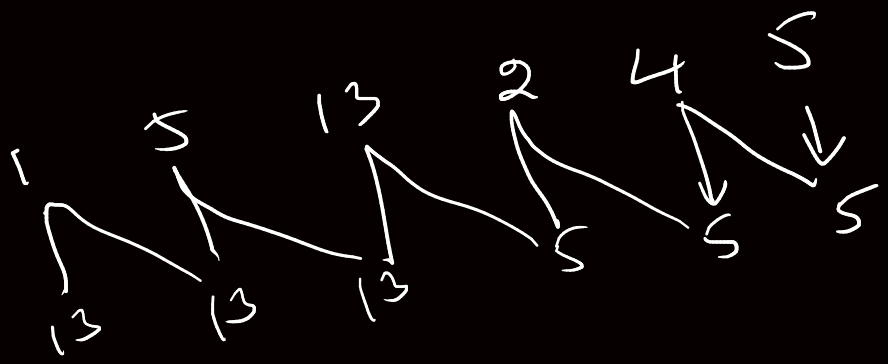
[1 5 13 2 4 5]

13 13 13 5 5 5

O(N²) → i−(i−1)

N*N-1

(N²−n)



1) Prefix Arrays / Cumulative / Sum Array
2) MAX / MIN Array
3) Kadane's Algo
   ↳ max sub array sum
4) stocks sell & buy.

arr [11 12 13 14 15]
(1) (2) (3) (4) (5)

Q 2

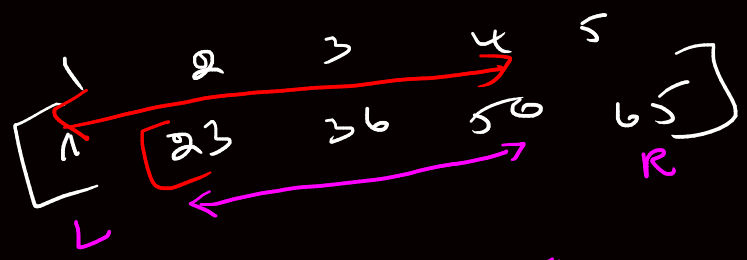
L R
2 4 → 12+13+14 → 39
1 5 → 11+12+13+14+15 = 65

arr [11 12 13 14 15]

pre [11 | 11+12 | 11+12+13 | 11+12+13+14 | 11+12+13+14+15]

pre [11 23 36 50 65]
↑

ans[3] pre[1-1]
12 + 11 = 23



2
2 4
1 5

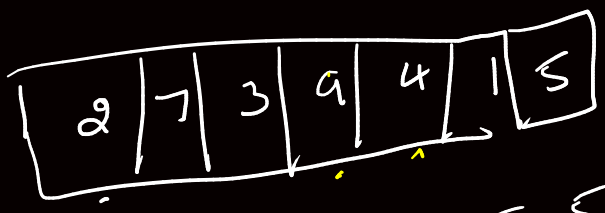
2
L 4 R

50 - 11 = 39

PFC4] - PFL2-1]

PFCR] - PFCL-1]

MAX ARR find maximum on right



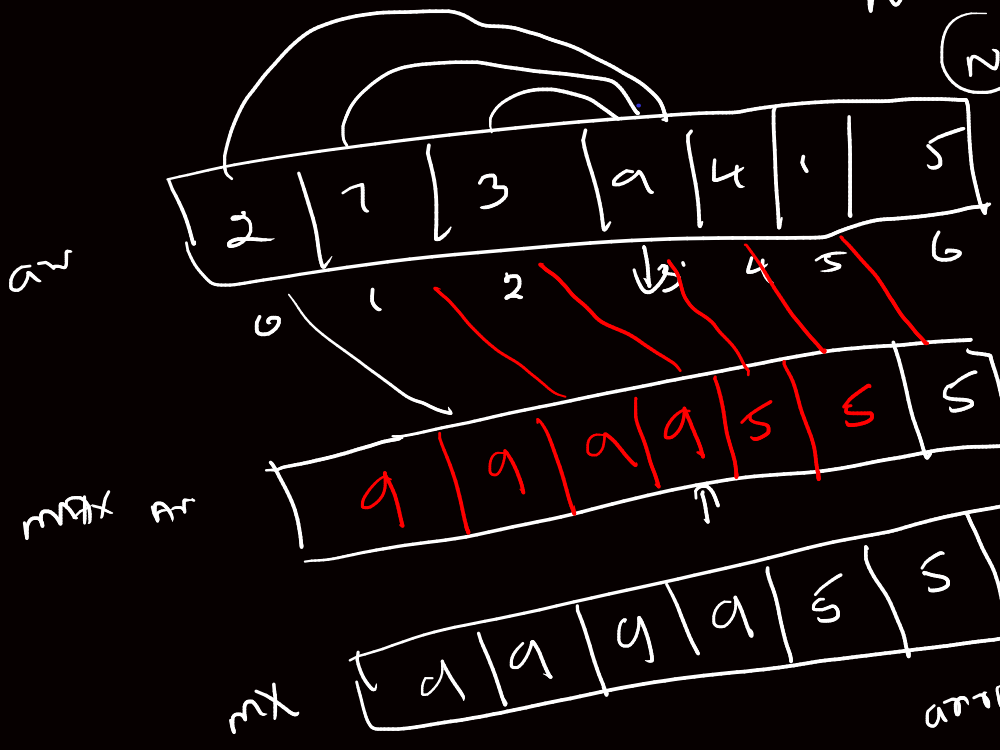
max = ~~4~~ 5

9 9 9 9 5 5 5

i ~~*~~ (i-1)

N * (N-1)

N² - N =



min Array

10 -221 -9 2 7 3 9 4 1 5

-221 -221 -9 1 1 1 1 1 1 5

DSA -1

Basic of prog

1) variable / Arrays
2) Loops / conditions
3) function
4) Basics of Recs.

Javac
JVM JDK
JRE

1)

"aba nanton

a → 3
b → 2
c →

2) Sorting ← Bubble
Insert
merge
QuickSort
Heap

3) Search → Linear → O(n)
Binary → O(log)

time ↓ space ↑

4) Hashing → O(1)

5) strings → replace
cout

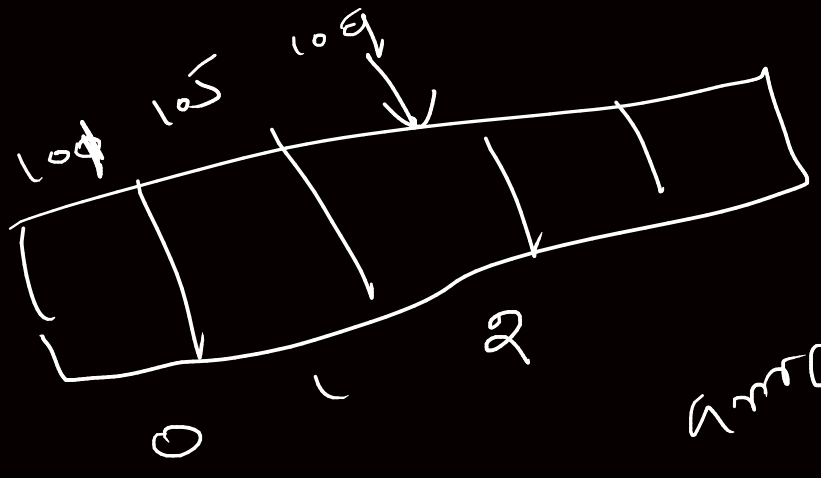
immutable
string
StringBuilder
Buffer

hash function

array

Base

100



Address

arr[2]

↓

100 + (2*4)

100 + 8 = 108

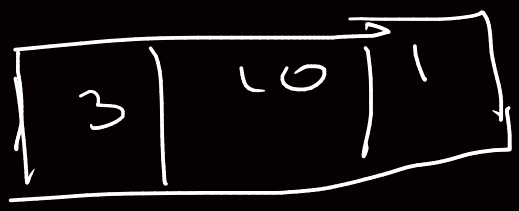
arr[0] = arr2

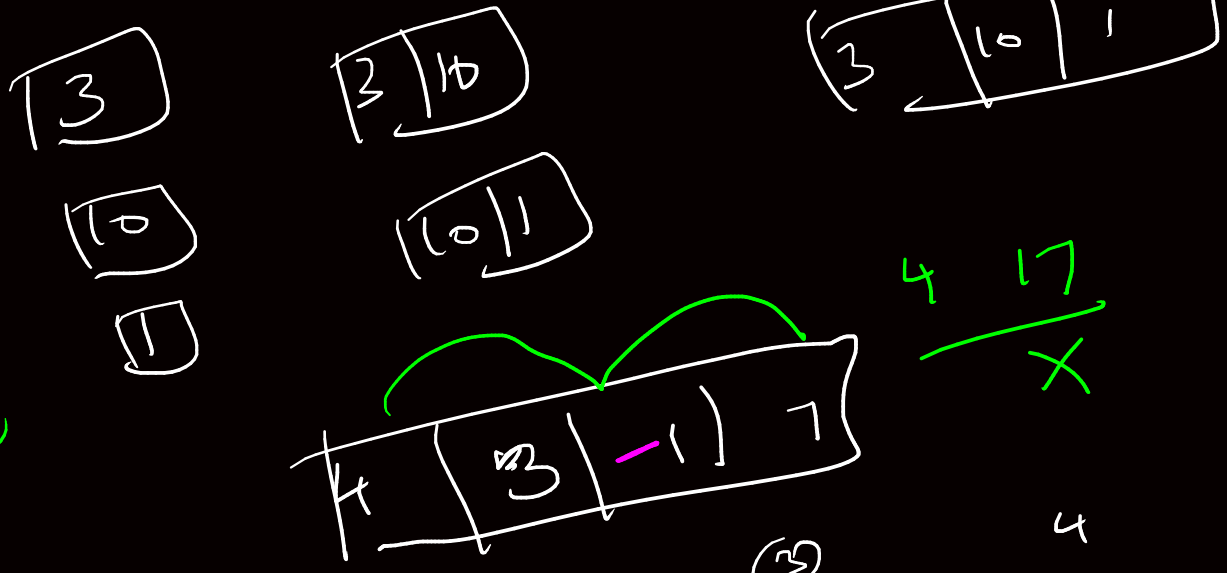
0 (6) 0 1

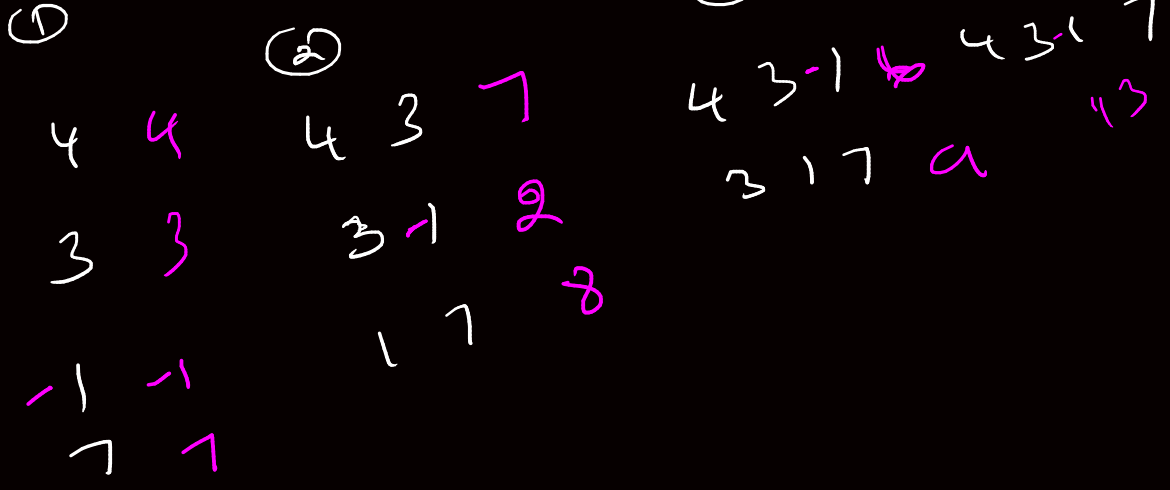
0 0 0 1 1 1

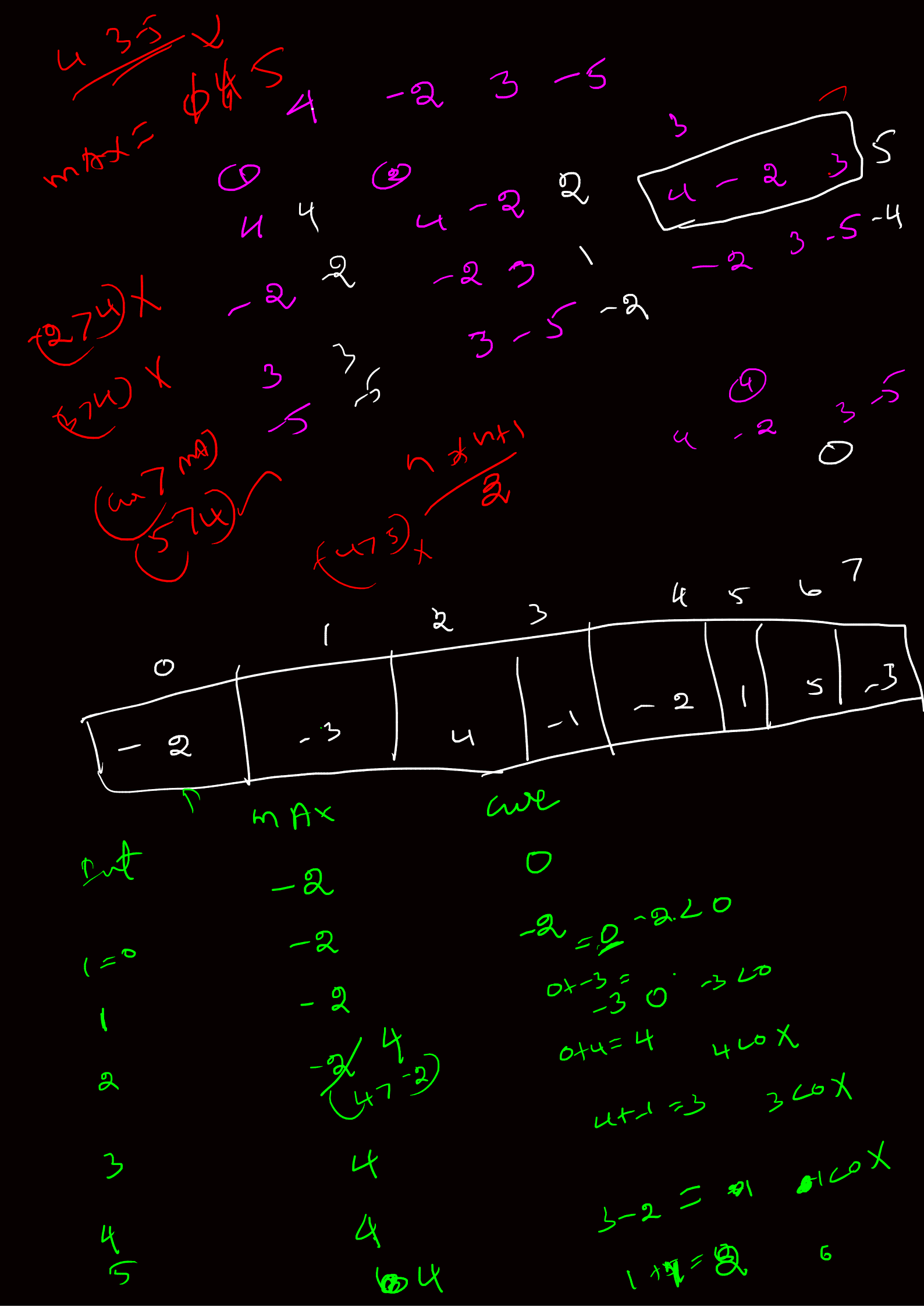
Kadane's Algo

Longest subArray with maximum sum








4 -2 3 -5
max=
0 1 2 3
4 -2 3 5
4 4 4 -2 2
-2 2 -2 3 1
-2 3 -5 -4
3 -5 -2
3 3 5
5
0 1 2 3 4 5 6 7
-2 -3 4 -1 -2 1 5 -3
max
cur
i=0
1
2
3
4
5
-2
-2
-2
-2 4
4
4
4
0
-2 = 0 -2 < 0
0+-3 = -3 0 -3 < 0
0+4 = 4 4 < 0 X
4+-1 = 3 3 < 0 X
3-2 = 1 1 < 0 X
1 + 1 = 2
6

6

7

(2+5)=7

7

(7)

2+5=7

7-3=4