

Bonus Class - Bit Manipulation

Special class

 Bitwise

Operators:-

`int a = 0`

`a` → 1

`-1` → hour

a	b	c/d
0	0	0
0	1	0
1	0	0
1	1	1

a	b	c OR o/f
0	0	0
0	1	1
1	0	1
1	1	1

A hand-drawn diagram illustrating a stack-based memory model. A large oval at the top represents memory, containing symbols like !, -, and 0. An arrow points from this oval down to a stack pointer (star symbol). The stack pointer points to a stack of values: 2, 1, and 2. To the left of the stack, curly braces indicate a block of memory. Below the stack, a horizontal line with arrows at both ends represents the heap.

The diagram illustrates a circuit connection. A central rectangular box contains a vertical column of four circles. The top circle is labeled 'a' above it and 'b' below it. The middle circle is labeled '1' below it. The bottom circle is labeled 'c' above it and 'd' below it. To the right of this central box is another vertical column of three circles, labeled '1', '2', and '3' from top to bottom. Arrows point from the top circle of the central box to the top circle of the right column, and from the middle circle of the central box to the middle circle of the right column. Another arrow points from the bottom circle of the central box to the bottom circle of the right column. To the right of the right column is a large circle containing the number '10'.

Decimal \rightarrow Binary =

Binary \rightarrow Decimal = already

1's comp

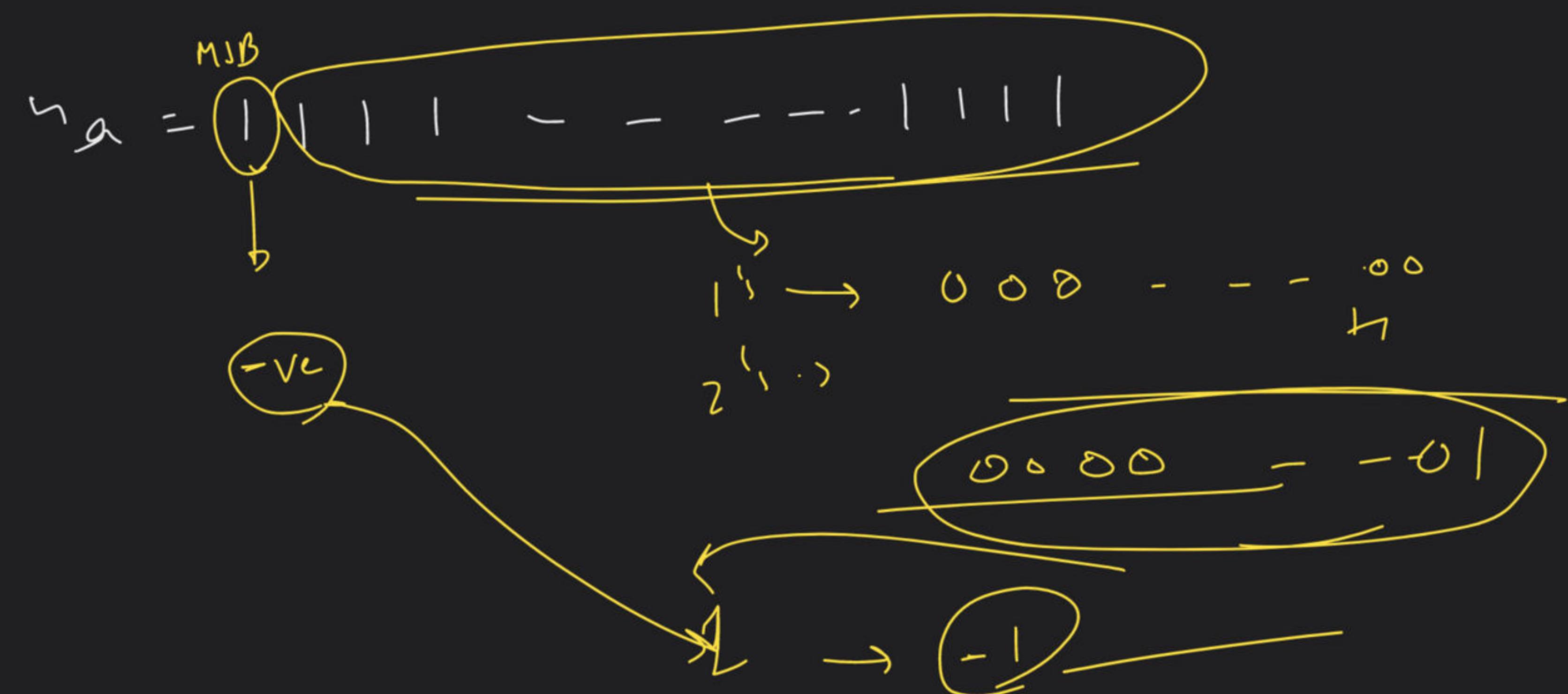
2's comp

sign

int a = 0

$$na = -1$$

$$a_2 \quad 000 - - - - - \quad 0000$$



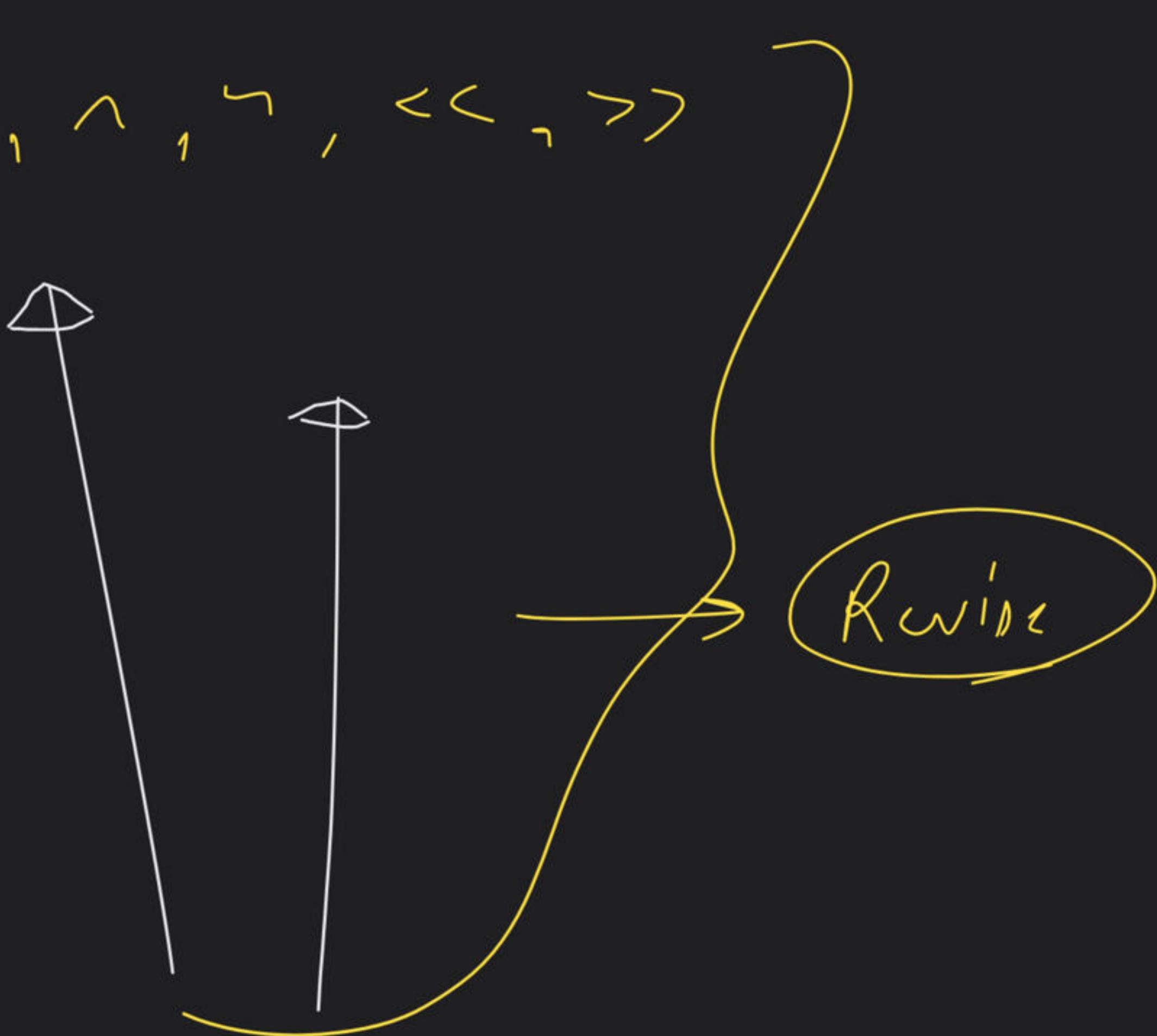
→ Bitwise op → & , | , ^ , ~ , << , >>

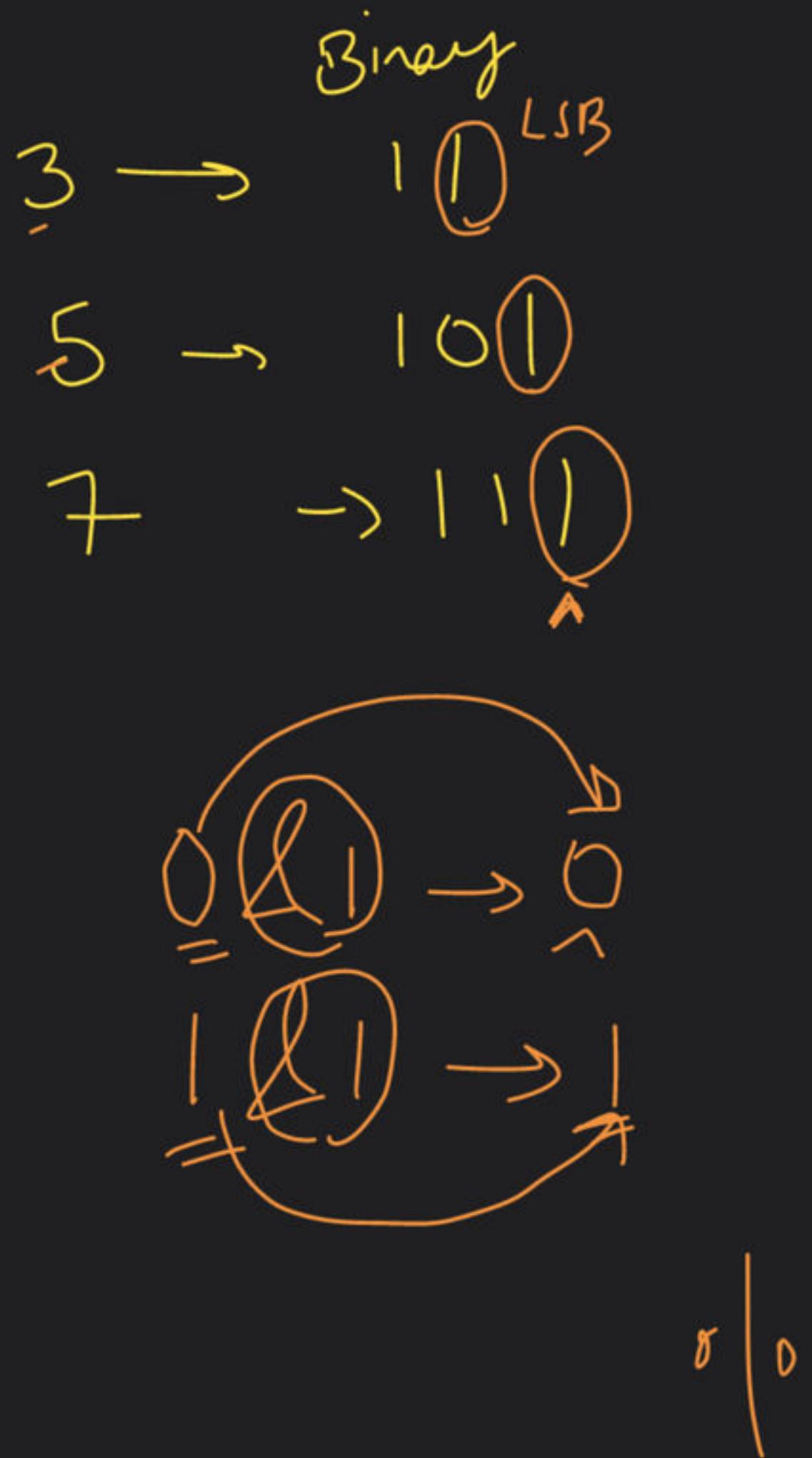
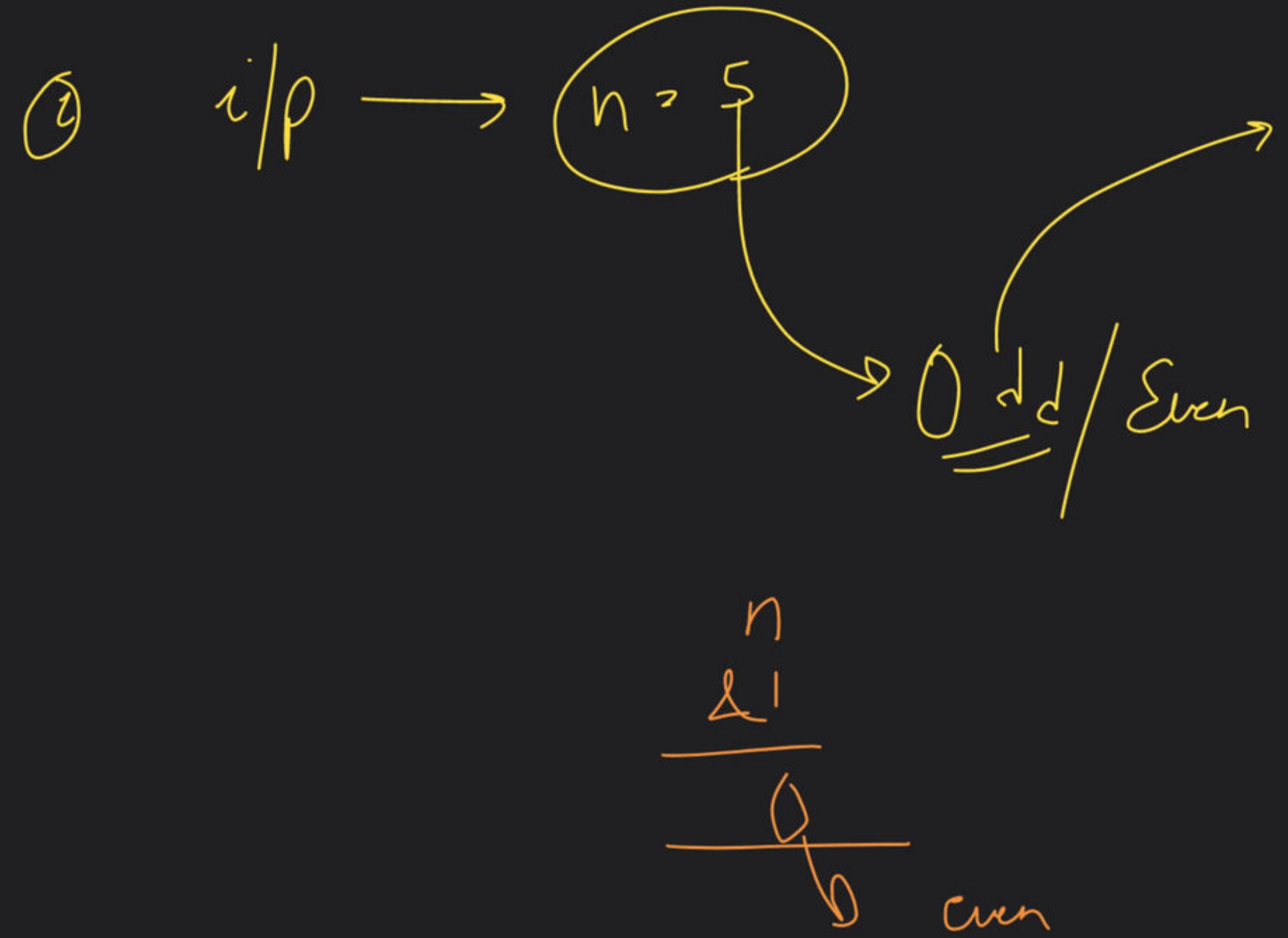
→ sign bit
} is comp

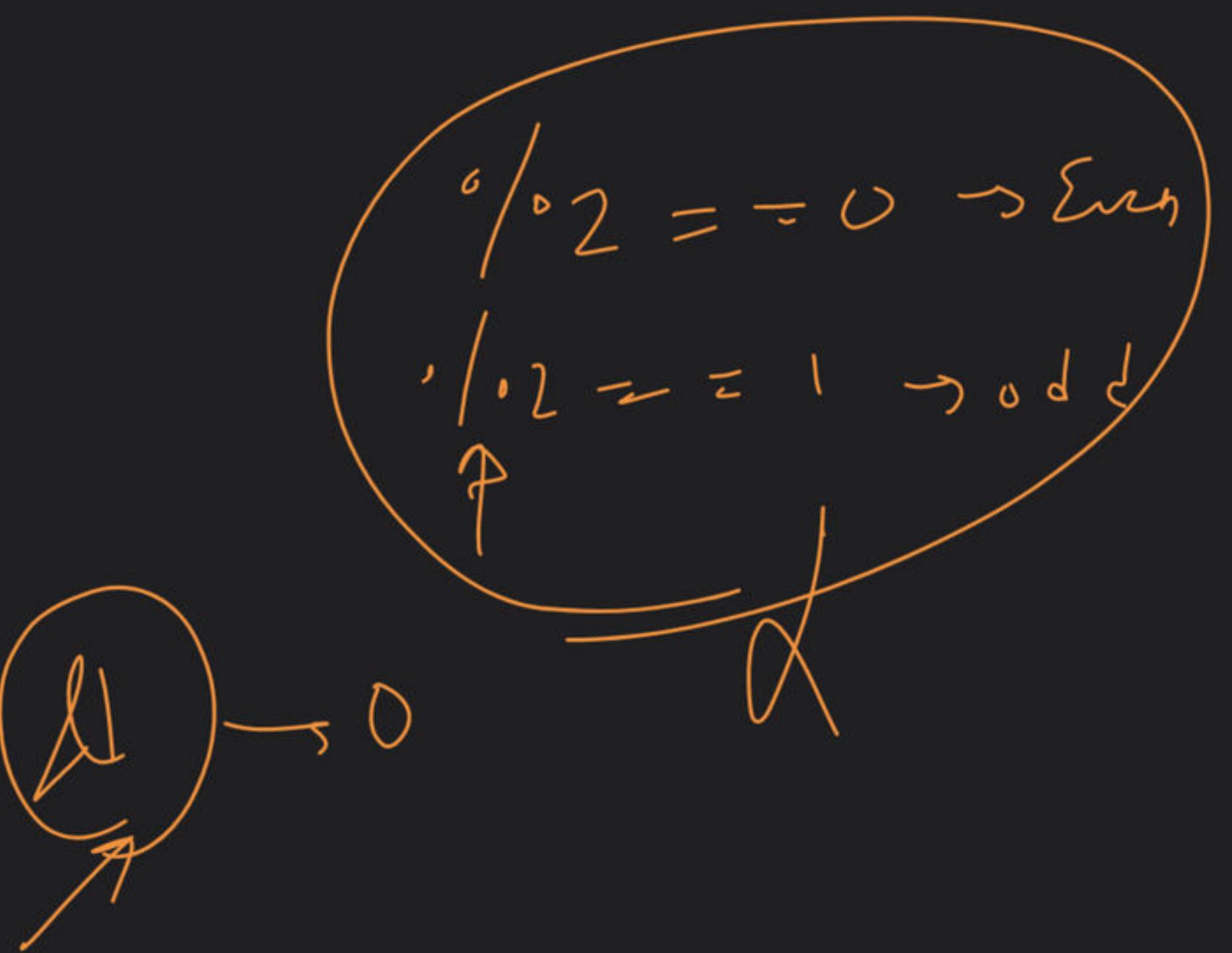
→ 2^b comp

→ D → B

→ B → D





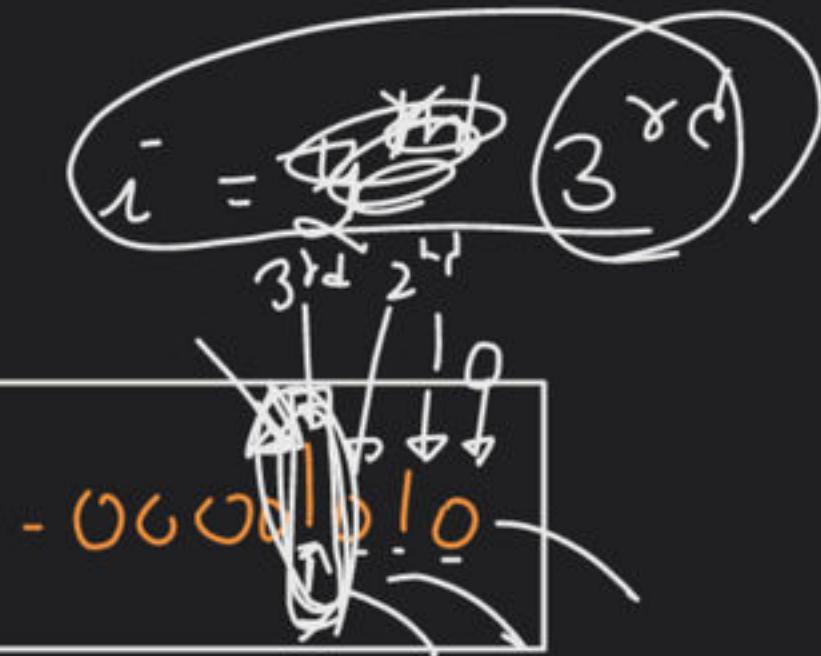


\rightarrow Mask

get i^{th} bit

int mark = $1 \ll i$
int ans = $h \& \text{mark}$

$\rightarrow h = 10$



LBB

$0 \& 1$

mask $\rightarrow 1$

0000 - - - 1

0000001000

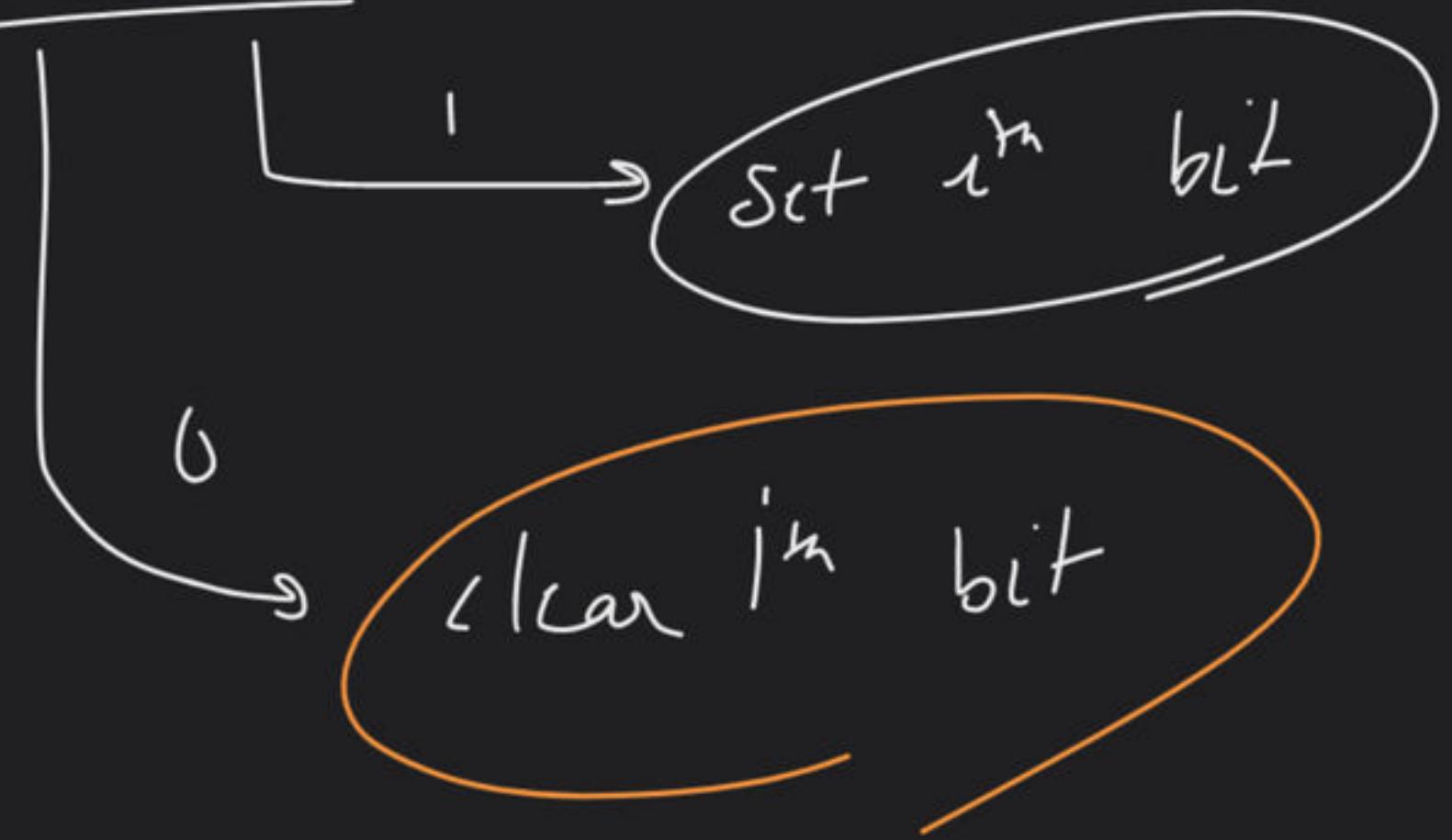
$\ll 3$

$\& \text{mask}$

$\rightarrow 0 \rightarrow \text{bit } 0$

000 $\rightarrow \text{bit } 1$

change i^{th} bit



Select i^{th} bit

$$n = 1010$$

$$i = 2$$

2nd bit loc act
by do

$\overline{1} \leftarrow 0^k$

$$\begin{array}{r} 0000 \\ - \\ 0000 | 0100 \end{array}$$

mask

$$0000\ 00$$

$$0000\ 0100$$

OR

$$\begin{array}{r} 00000000 \\ - \\ 00001110 \end{array}$$

0
1
1

break mask
mask $\leftarrow 1 \ll i$

$1 \ll 2$
 $1 \ll i$

$$0000 | \ll 2$$

$$0000100$$

$au = n | . mask$

Clear its bit

\neg

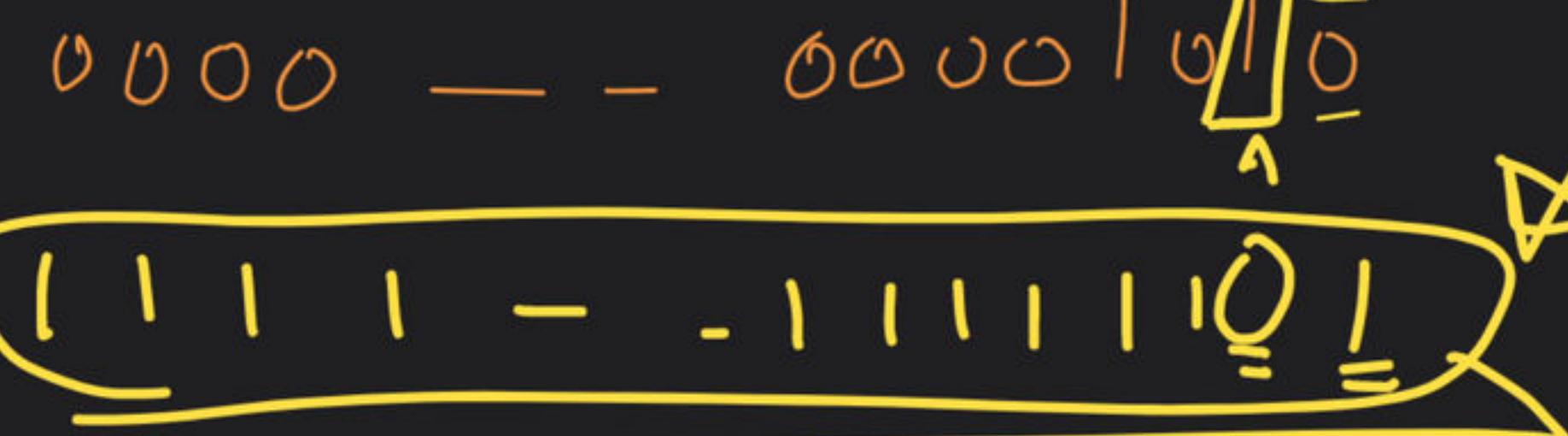
$n = 10$

$i = 1$

\neg

1

0



0000 0 0000 1000

(8)

set

011

$\neg(0000 \dots 0010)$

$\neg(i < i)$

Pata lagana

$\text{target} = 1$

mask

$1 < (1)$

$n=10$

Update i^{th} bit

01

1 0 1 0
0 1 0 0

1 1 0

$2^{n-2} \text{ } 2^{n-1} \text{ } 0^n$

1 0 1 0

0 1 0 0

1 1 0

act
dear
 $\text{for}(n, i, \text{target})$
 $0/1$

i
 $2^{\text{nd}} \text{ bit} \rightarrow \text{target} = 0$
 $i^{\text{th}} \text{ bit} \rightarrow \text{target} = 1$

1 0 1 0
0 0 0 0

1 0 1 0

1 0 0 0

1 0 0 0

11
00

Update i^{th} bit

①

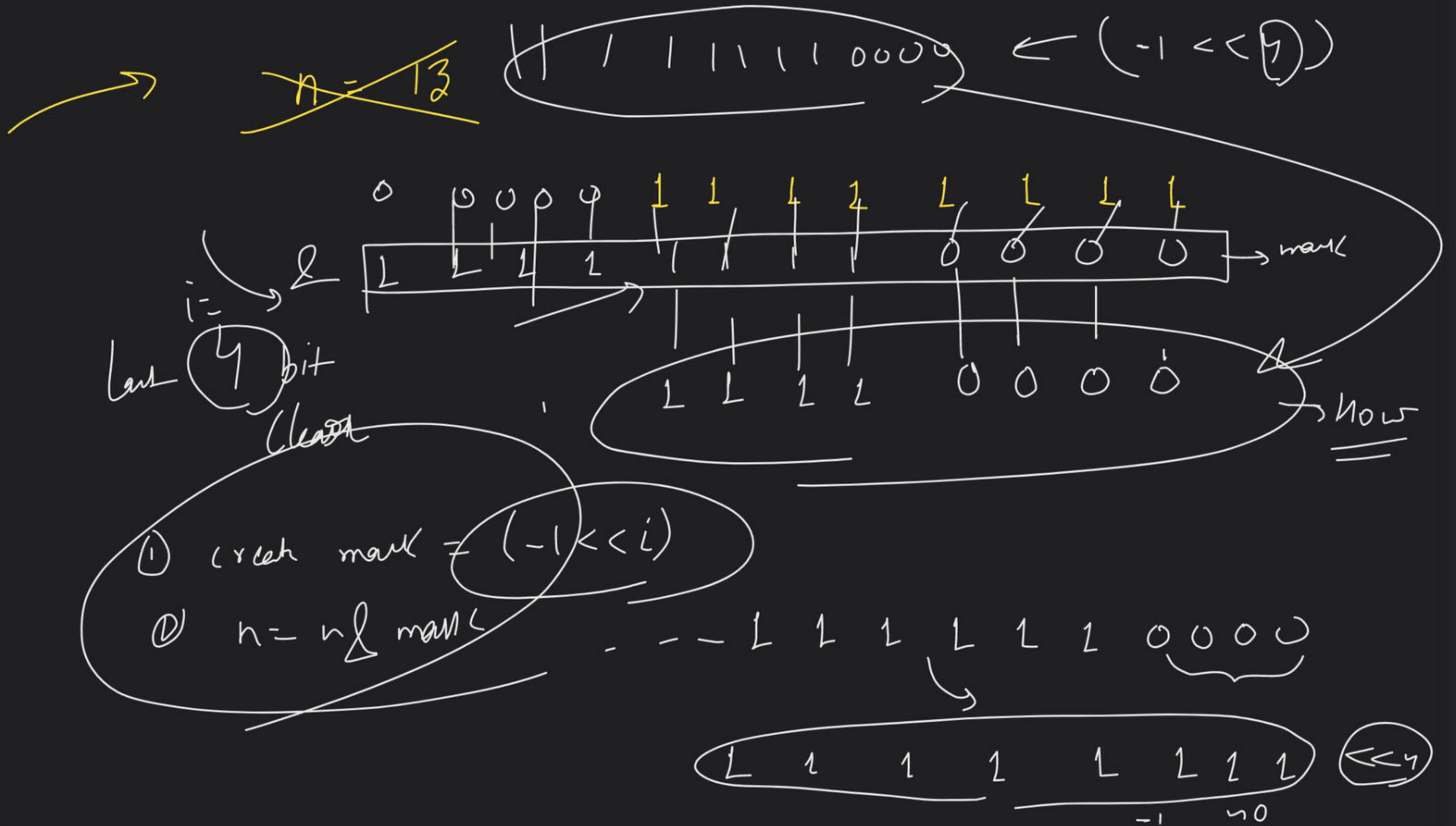
clear i^{th} bit

mask \rightarrow

target $<< i$

$n | \text{mask}$

$$\text{Y} \neq \text{Z} \wedge \text{Z} \sim \parallel \wedge \Delta$$
$$= \text{Y}$$



00000 - . 600 

0000 - - 600 

(1)



Check

power of 2

$$\left(\frac{16}{2} \rightarrow \frac{8}{2} = 4 \right)$$

$\times 2$

if $\rightarrow n = 16$

2, 4, 8, 16, 32, 64

128, 256

512, 1024

$\times 2$

$\times 2$

$$8 \rightarrow 1000$$

$$7 \rightarrow \begin{array}{r} 0111 \\ \hline 0000 \end{array}$$

$$2^1 \rightarrow 2 \rightarrow \begin{array}{r} 10 \\ \hline 100 \end{array}$$

$$2^2 \rightarrow 4 \rightarrow \begin{array}{r} 100 \\ \hline 1000 \end{array}$$

$$2^3 \rightarrow 8 \rightarrow \begin{array}{r} 1000 \\ \hline 10000 \end{array}$$

$$2^4 \rightarrow 16 \rightarrow \begin{array}{r} 10000 \\ \hline 100000 \end{array}$$

$$16 \rightarrow 100000$$

$$15 \rightarrow \begin{array}{r} 01111 \\ \hline 00000 \end{array}$$

$$n \rightarrow \begin{array}{r} 1 \\ \hline 011 \end{array} \rightarrow \begin{array}{r} 100 \\ \hline 000 \end{array}$$

$$n-1 \rightarrow \begin{array}{r} 3 \\ \hline 000 \end{array}$$

$n \wedge (n-1) = 0$

if it is a power of 2

Count
8th bit

Count set bits



8 → 0000 - - - 00 [1]000

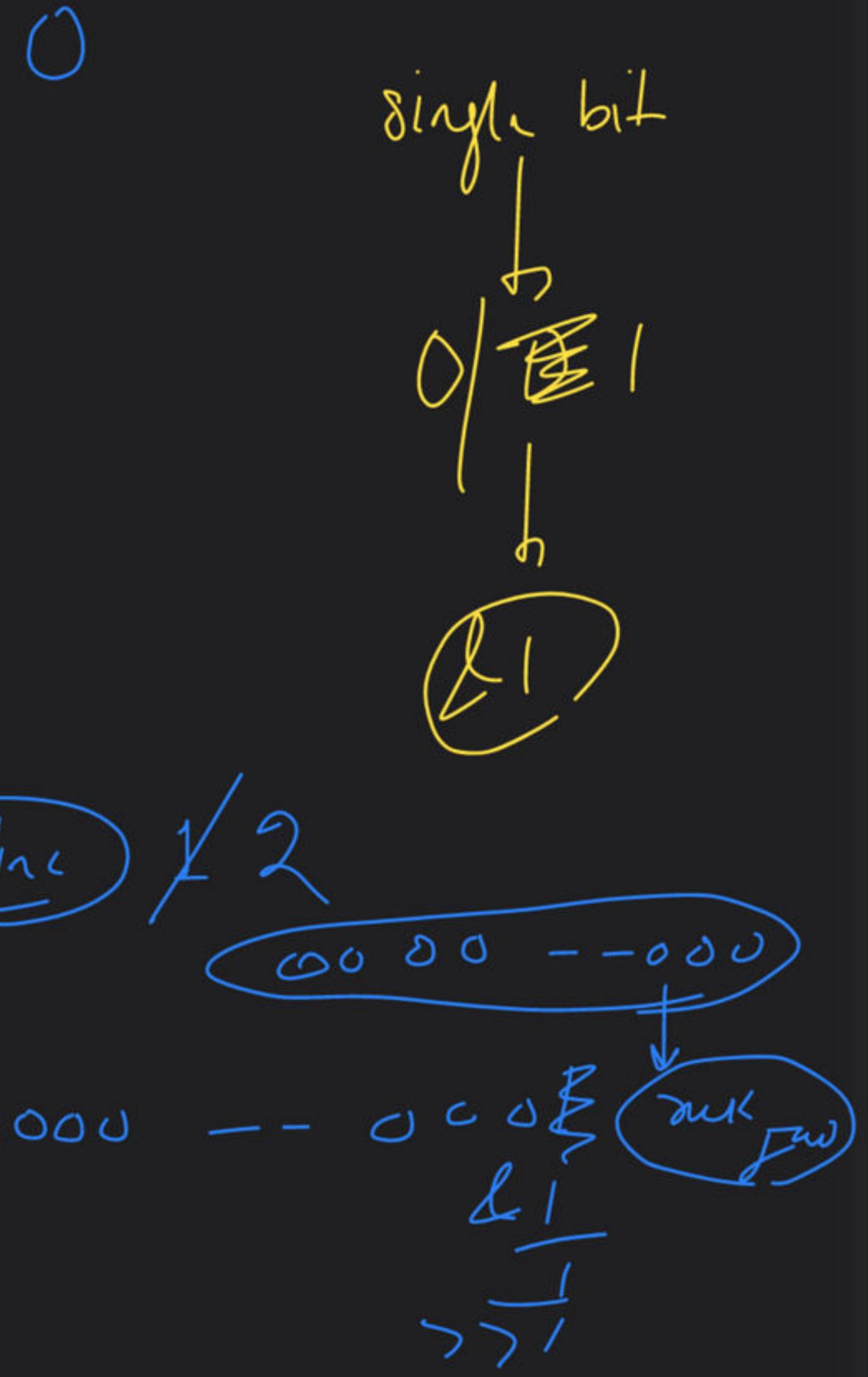
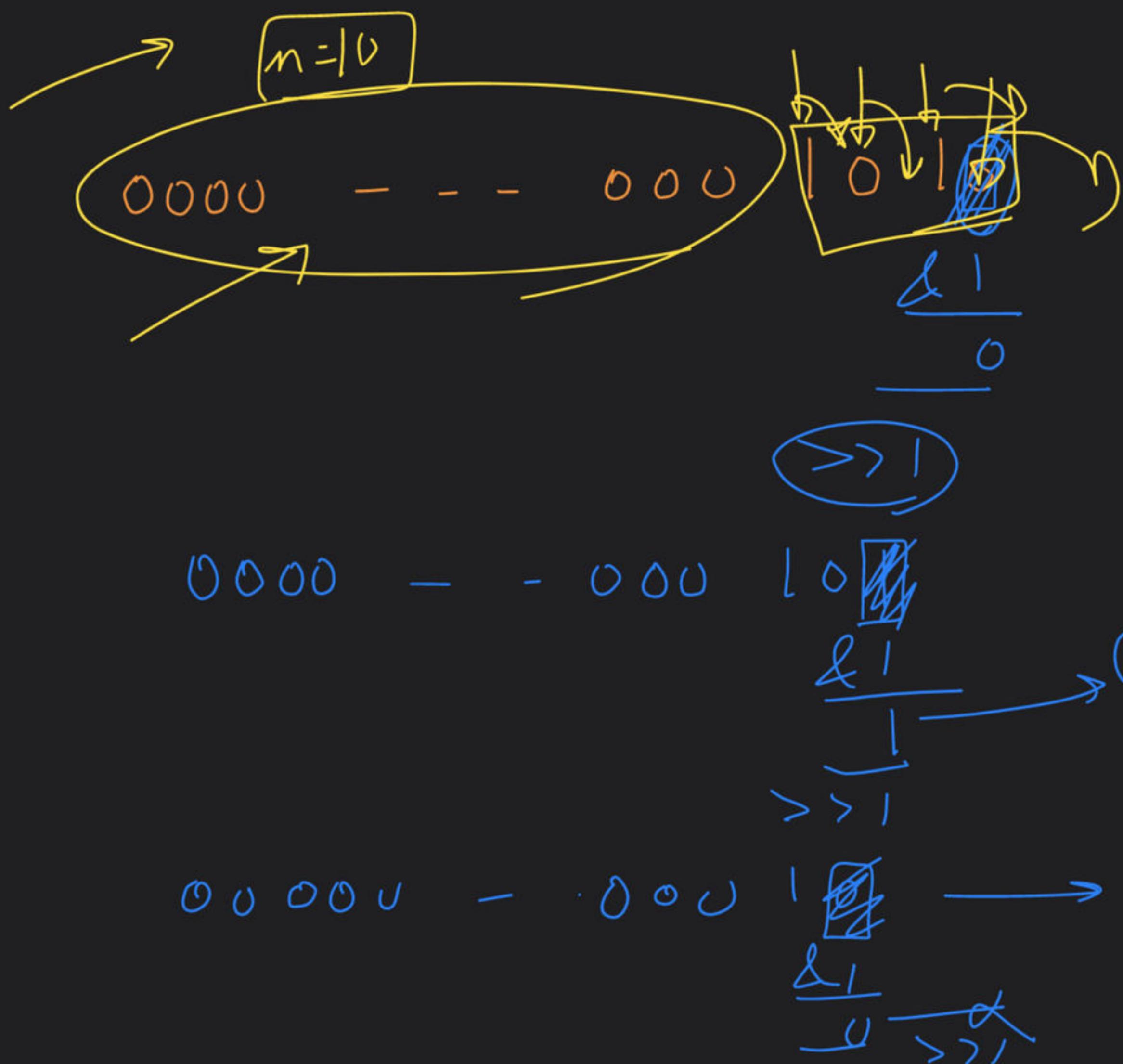
no. of set bit = 1

7 → 000001 - 000 [111]

no. of set bit = 3

10 → 000000 - - - [1010]

no. of set bit = 2



far +

6 0 0 0

10

10

1

○ △

20

U

Fast Exponentiation

$$n \rightarrow \text{even/odd}$$

$$a^n$$

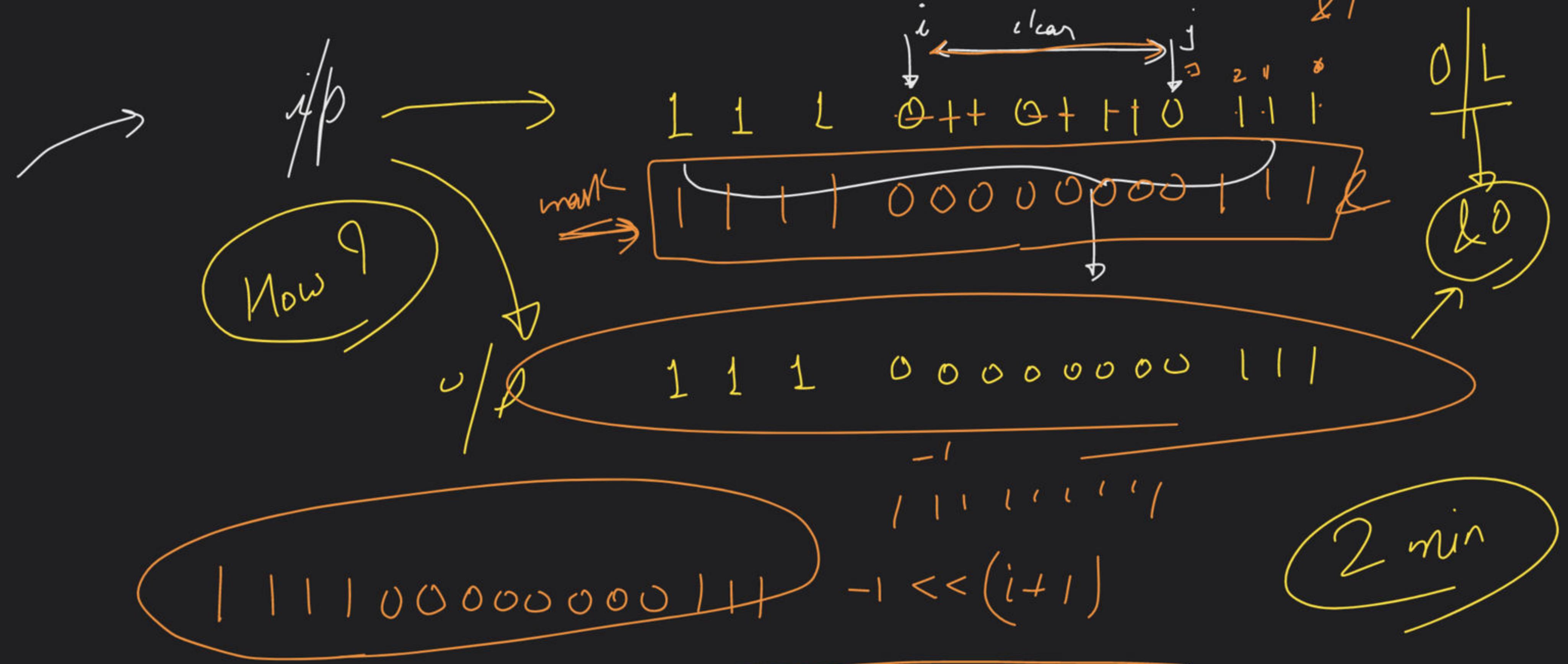
$$n \rightarrow \text{even}$$

$$a^{n/2}$$

$$n \rightarrow \text{odd}$$

$$a^{(n-1)/2} \cdot a$$

$$\alpha \rightarrow \star \alpha$$



$$a = (-1 << (i+1))$$

$$b = (L << j) - 1$$

$a | b$



$00000000000000000000000000000000$

$00000000000000000000000000000000$

OR

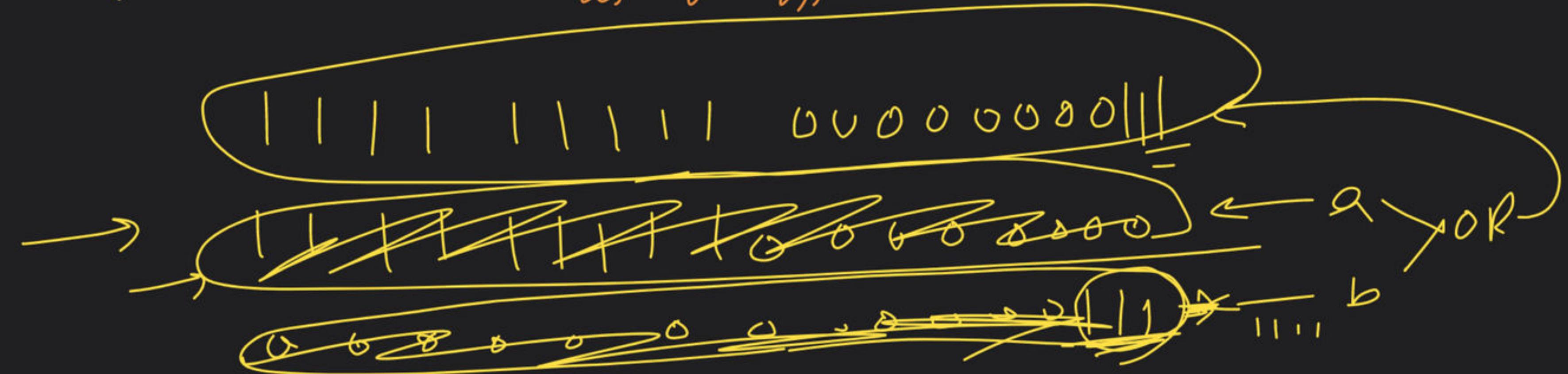
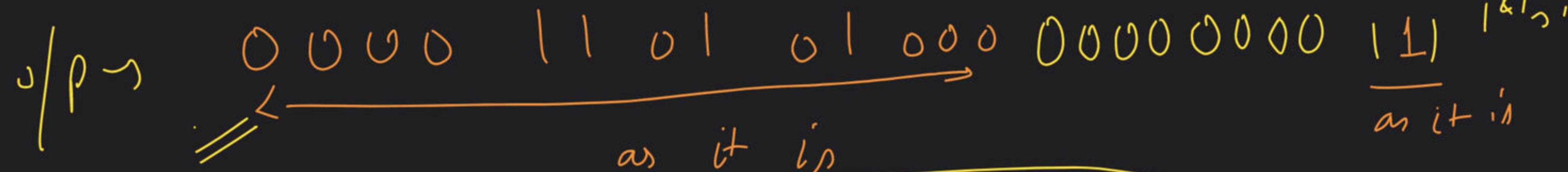
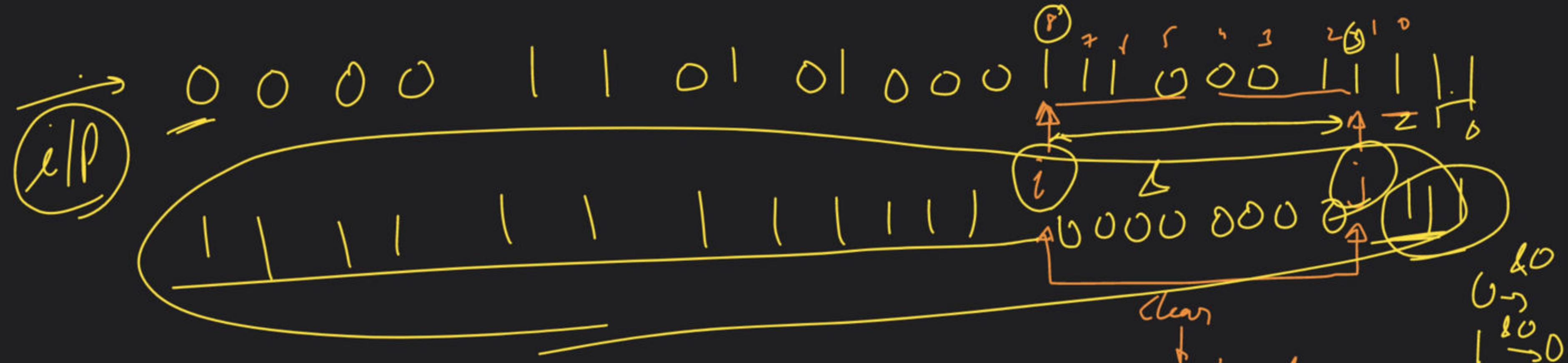
15

$$0000 - 0000 \begin{matrix} i \\ j \\ , \\ , \end{matrix}$$

$$0000 - 000 \overline{100}$$

\hookrightarrow eff

$f(15, 2, 1)$



A sequence of vertical bars and circles. The bars are positioned above the circles. Below the circles is a horizontal double-headed arrow labeled 'g' at both ends, indicating a cyclic or periodic pattern.


L.S.

$n-1$

$2 \rightarrow 10 \rightarrow 1$

$4 \rightarrow 100 \rightarrow 11$

$8 \rightarrow 1000 \rightarrow 111$

$16 \rightarrow 10000 \rightarrow 1111$

1111
 3^4

32

$\frac{1000}{-1}$

$\ll j$
 $\ll j$

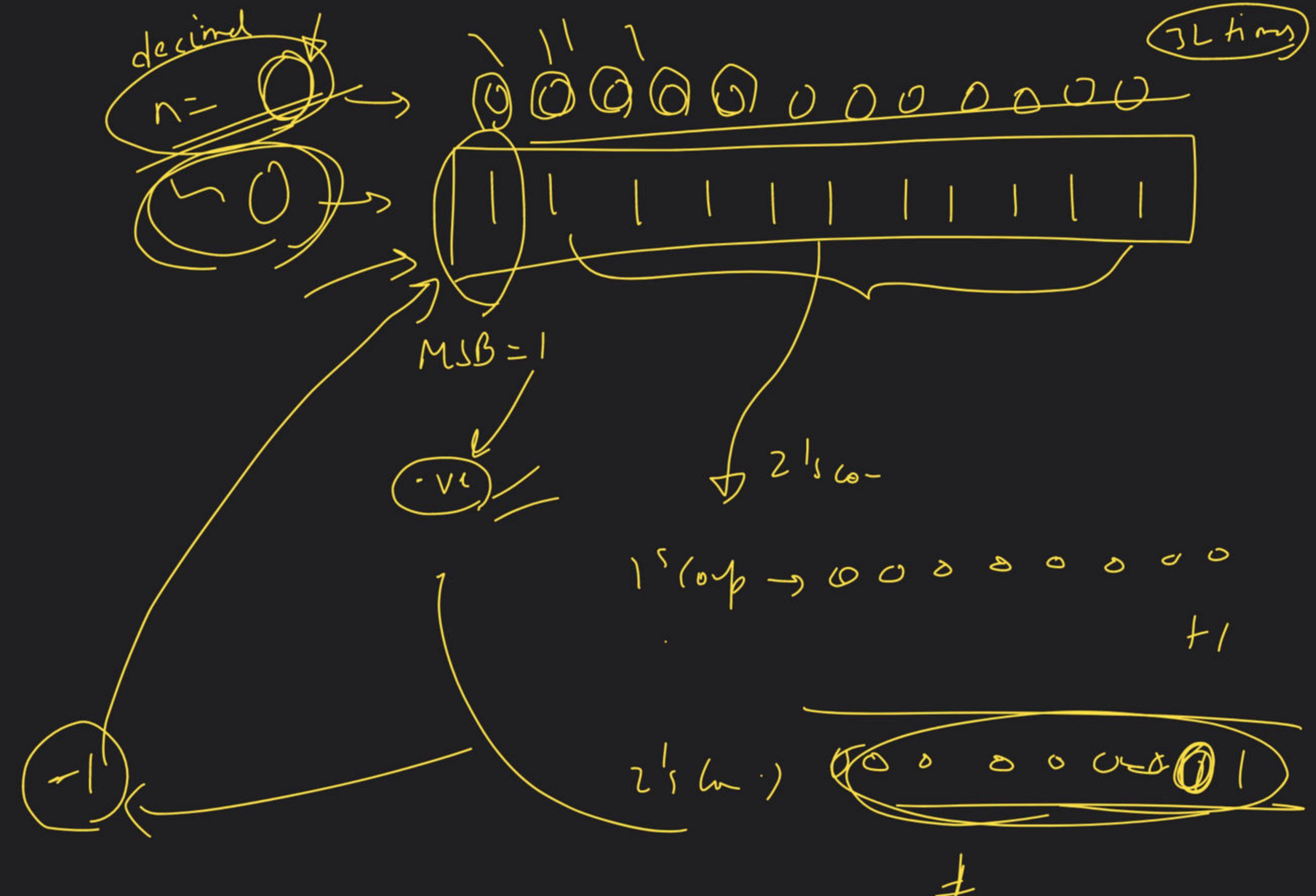
000000000

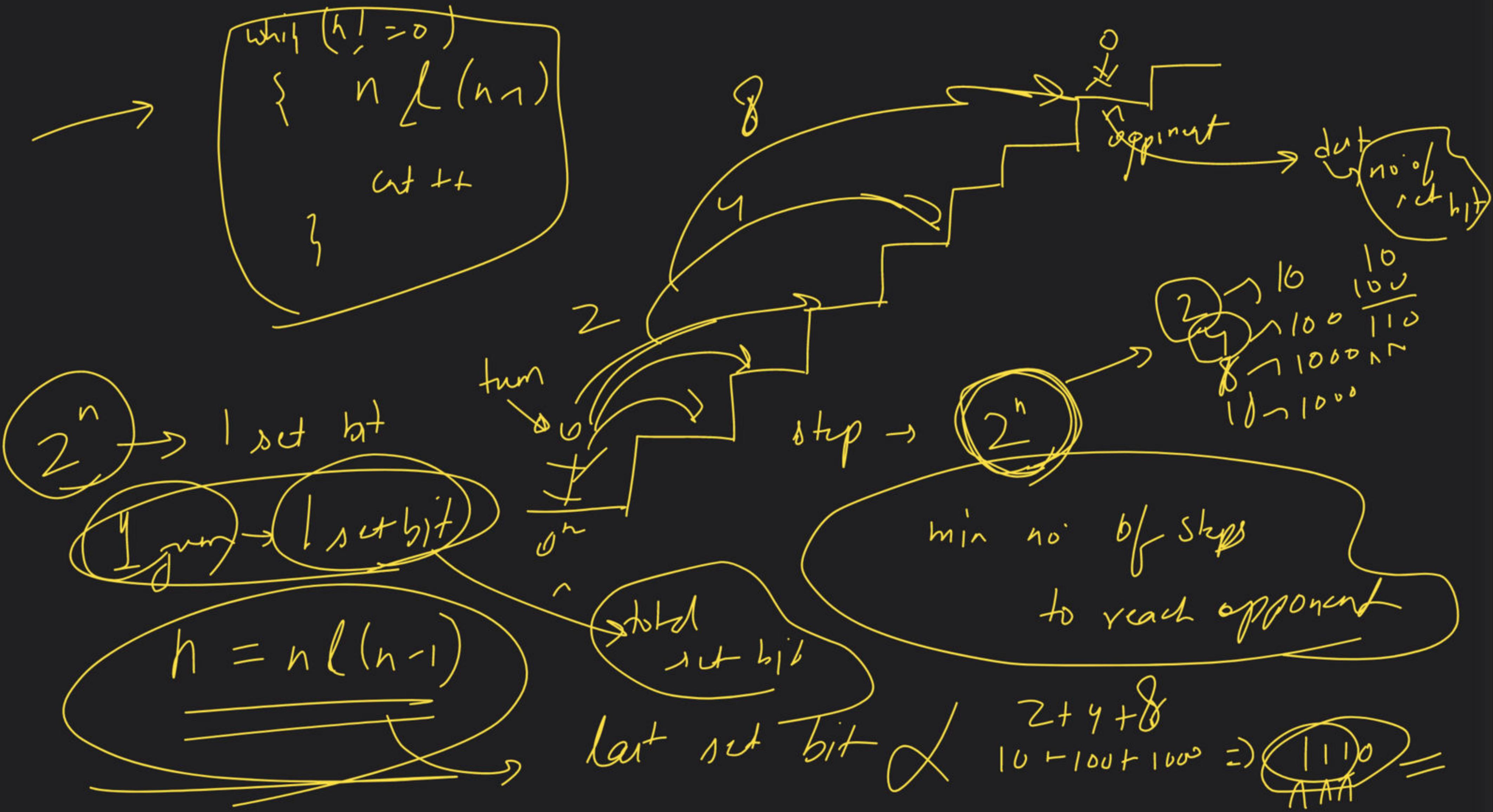
$(1 \ll j) - 1$
 $b \rightarrow$

000000000

000000000

a/b → mark





Subsequence Using Bitmasking

$n \rightarrow \text{length of str}$

$\text{str} = "abc"$ → Power set or Subsequence →

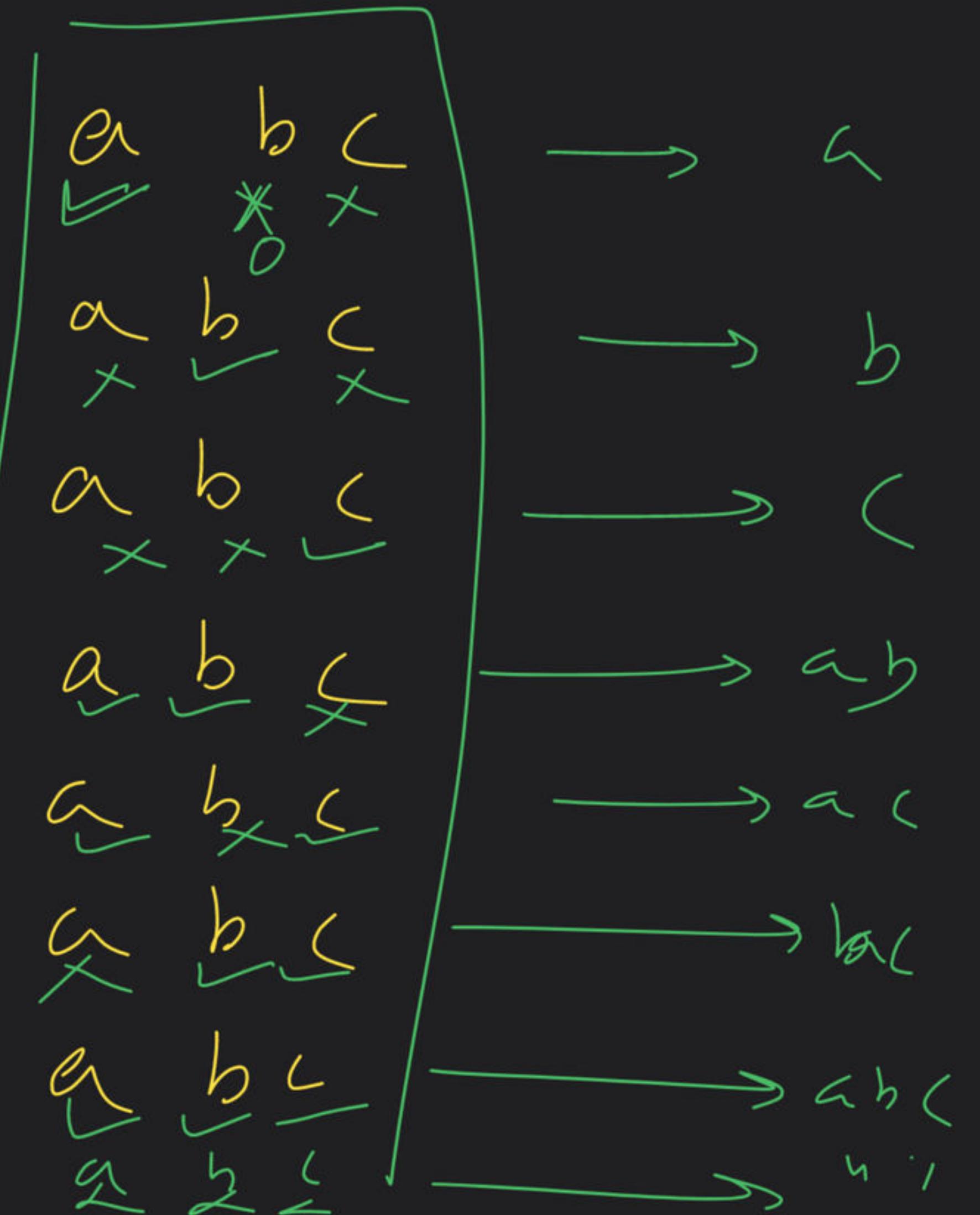
already solved → Revision → index pattern

Diagram showing all subsequences of "abc":

" "
a
b
c
ab
bc
ac
abc

$\Rightarrow 2^n$

L



"abc"

length $\rightarrow n$

$$n = 3$$

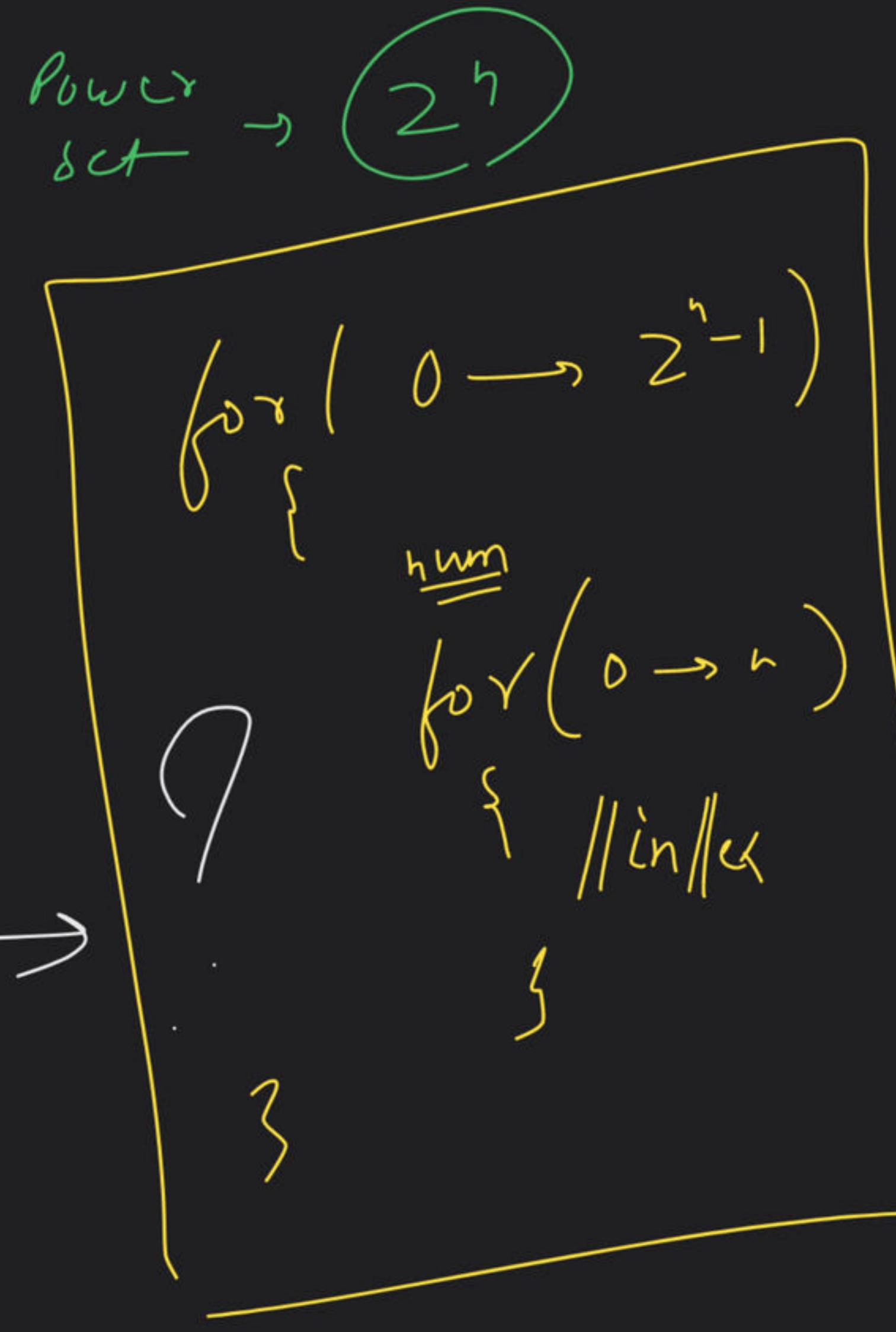
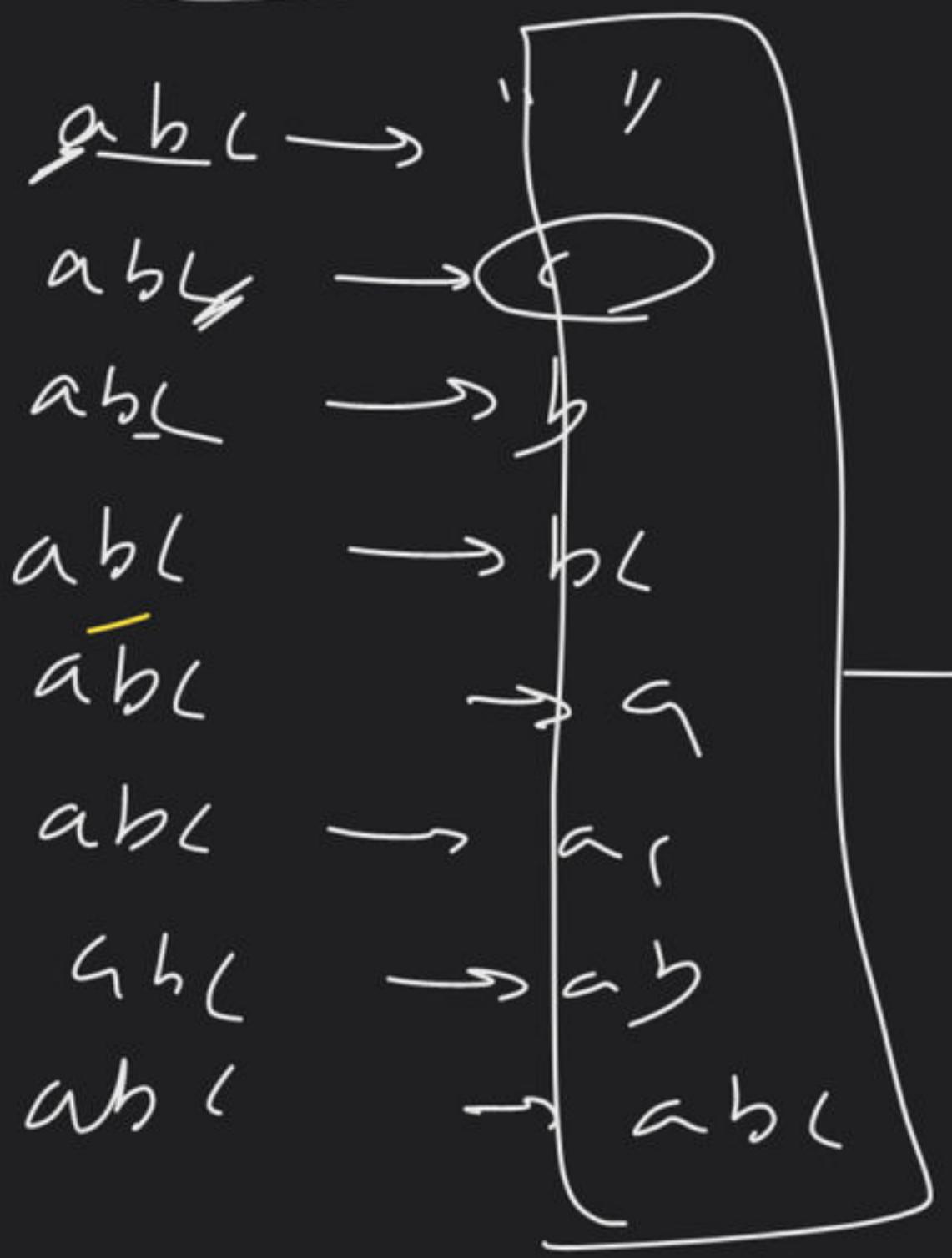
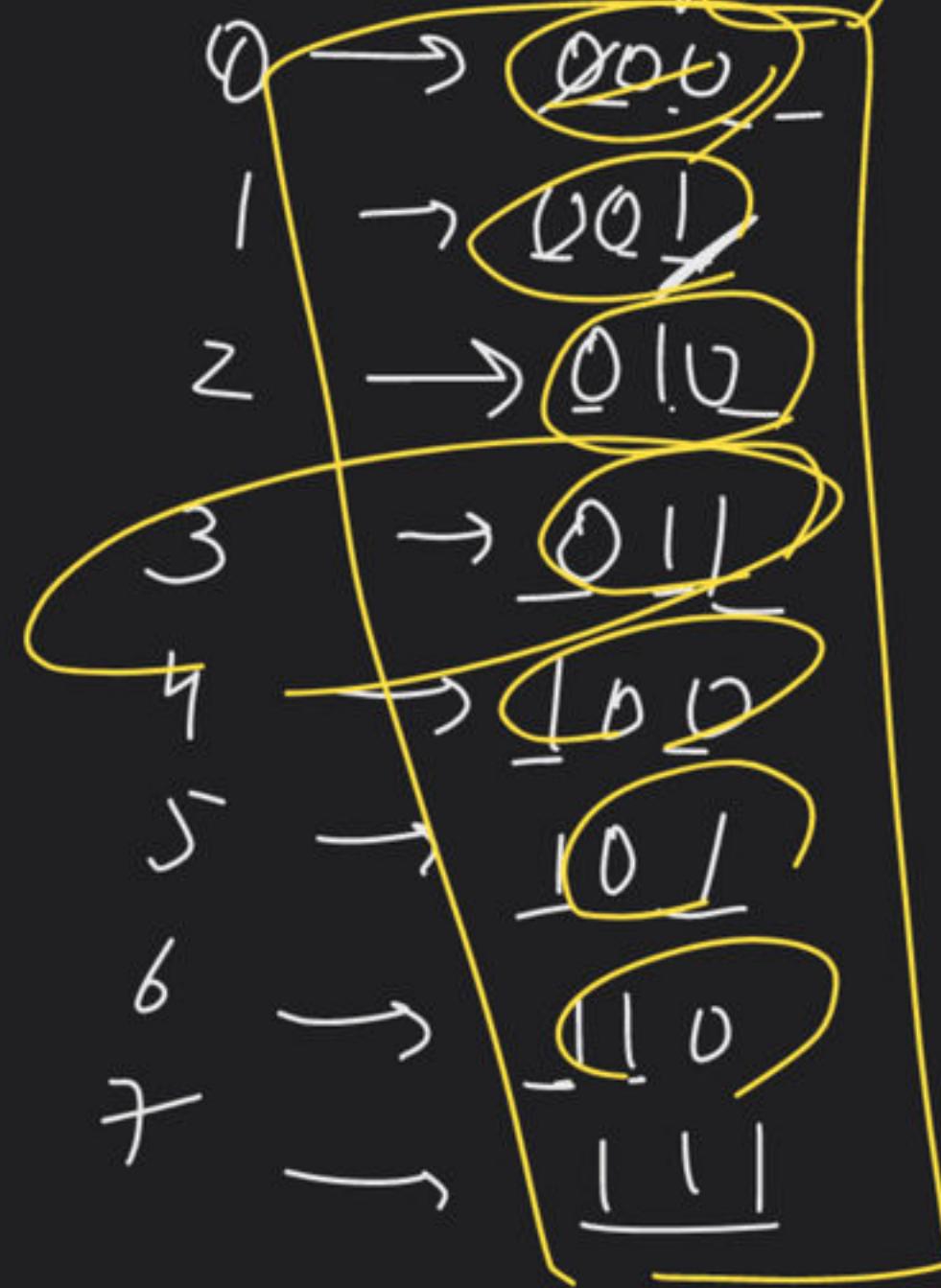
$$2^n = 2^3 = 8$$

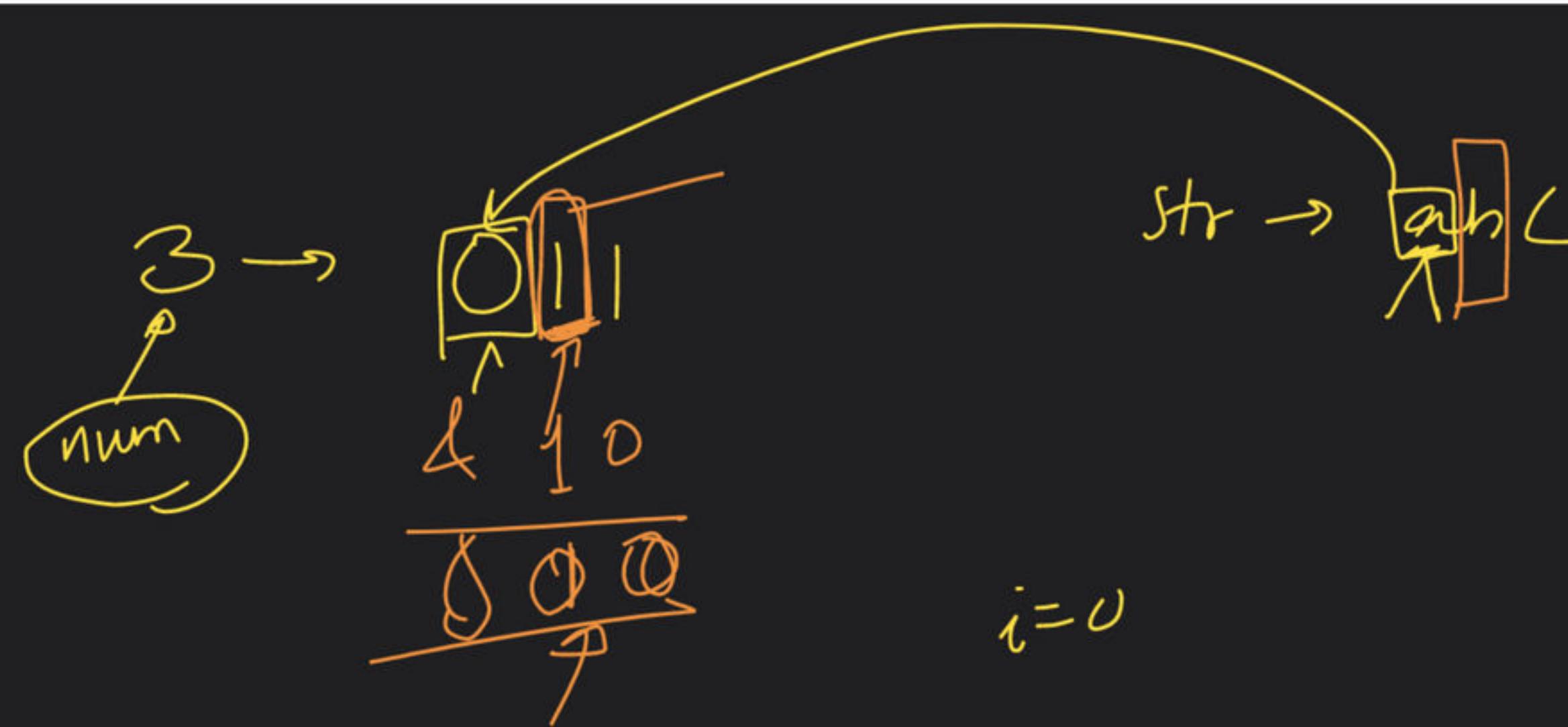
$0 \rightarrow 2^n - 1$

$0 \rightarrow 8 - 1$

$0 \rightarrow 7$

Binary





$i = 0$

$i = 1$

temp



$i = 2$

$i < < 2$

100

$i = 1$

$i < < 1$

010



$i = 0$

