

# Recursion - Level 3

Special class

Rc  
LS  
R.T  
F.R  
D.C

→ i/p → array

o/p → is Sorted or not Sorted

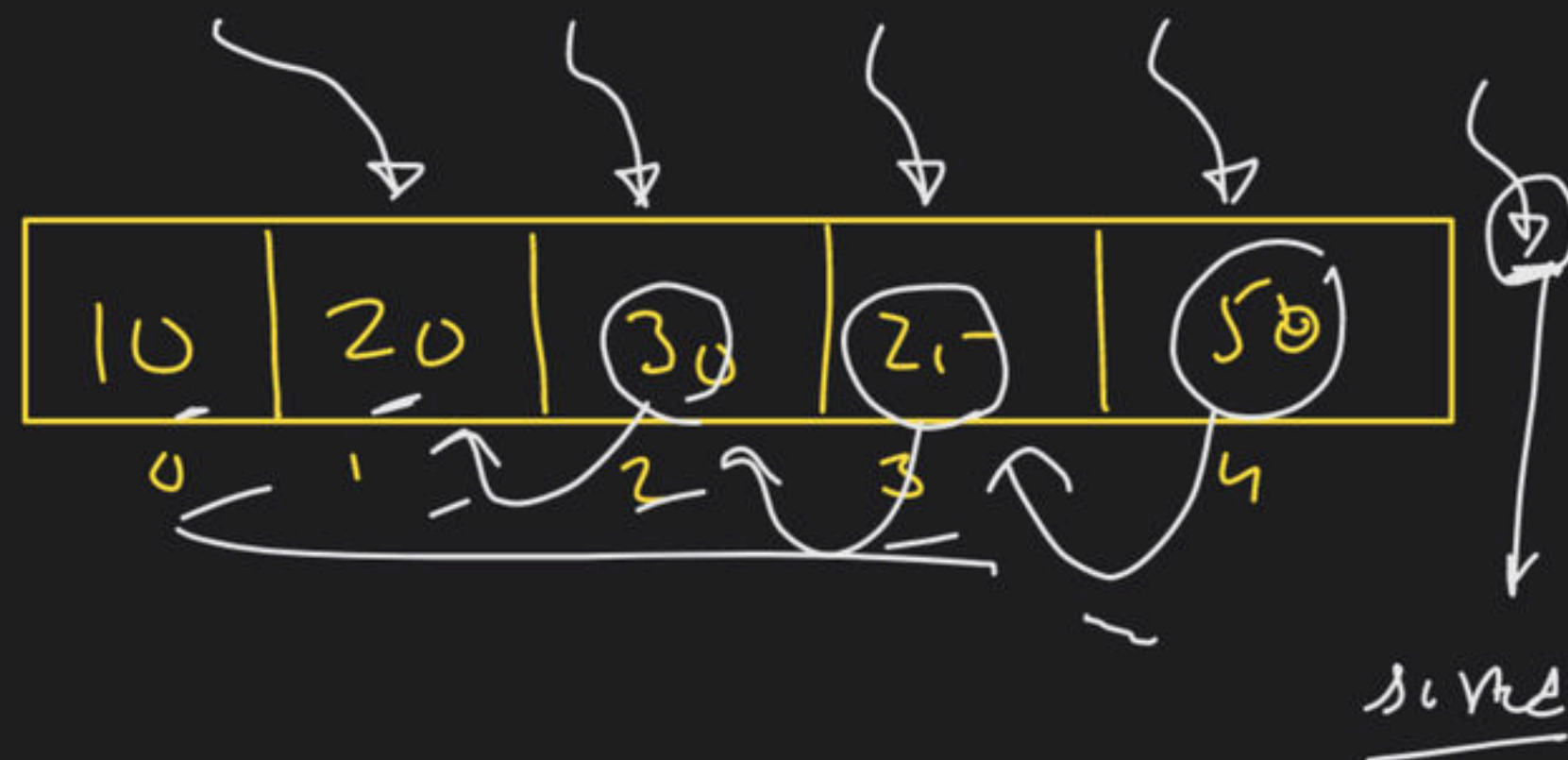
True / False

if (i == n)  
return

count < arr[i]

print(arr, n, i+1)

asc → a, b, c, d





not sorted

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 2 | 4 | 6 | 8 | 9 | 7 |
|---|---|---|---|---|---|

sorted

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 10 | 20 | 30 | 40 | 50 | 60 |
|----|----|----|----|----|----|

not sorted

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 10 | 20 | 40 | 30 | 60 | 50 |
|----|----|----|----|----|----|

0 1 2 3 4 5

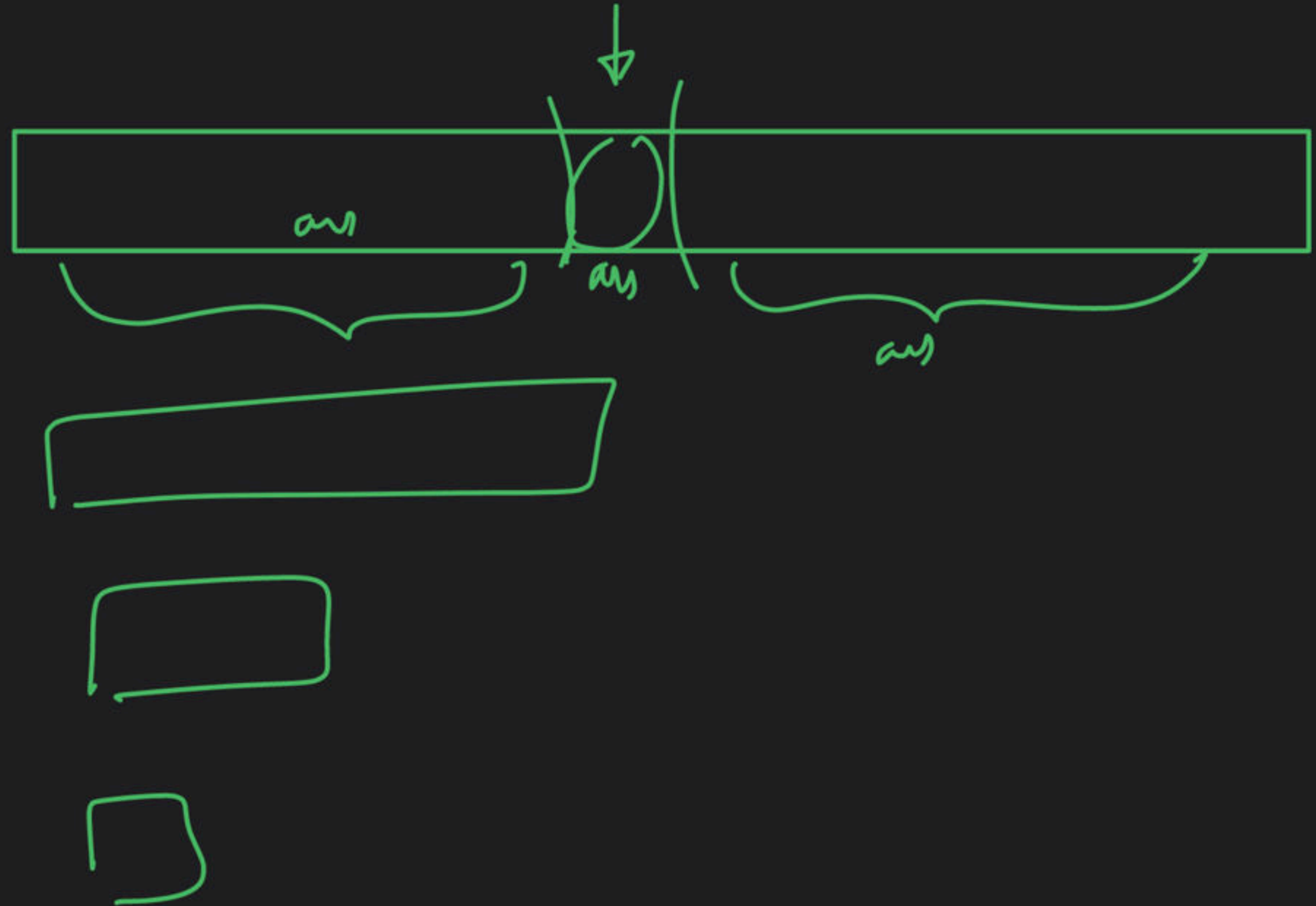
if  $arr[i+1] > arr[i]$   
→ aage bad ho

chr.  
return false

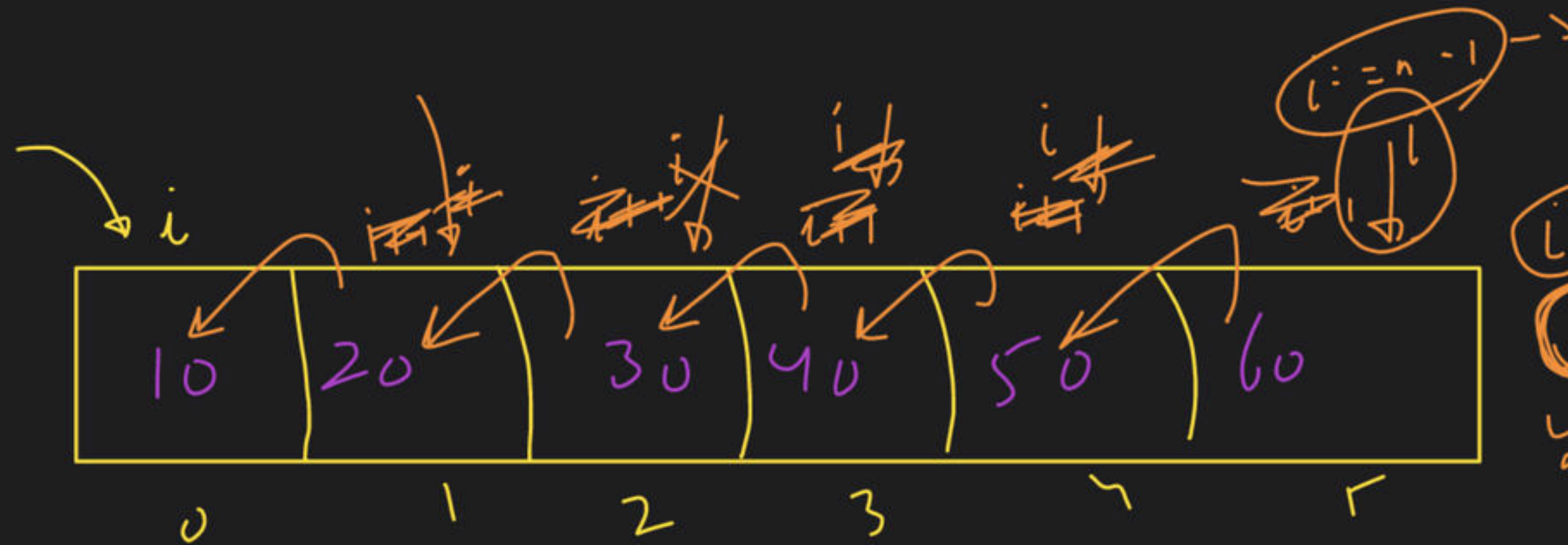
if  $i == n-1$   
→ return true

→ Binary Search:

$O(\log n)$







$(i \geq n)$   
 return true;

$(arr[i+1] < arr[i])$   
 return false;

→ check sort(arr, n, i+1)

|   |  |   |  |   |  |   |  |    |  |    |
|---|--|---|--|---|--|---|--|----|--|----|
| 2 |  | 6 |  | 7 |  | 9 |  | 12 |  | 11 |
|---|--|---|--|---|--|---|--|----|--|----|

|   |  |   |  |   |  |   |  |   |  |   |
|---|--|---|--|---|--|---|--|---|--|---|
| 4 |  | 3 |  | 1 |  | 2 |  | 4 |  | 4 |
|---|--|---|--|---|--|---|--|---|--|---|









# Binary search

```
int bs (arr, n)
```

```
{  
    int s = 0;
```

```
    int e = n - 1;
```

```
    int mid =  $\left(\frac{s+e}{2}\right);$ 
```

```
    while ( $s \leq e$ )
```

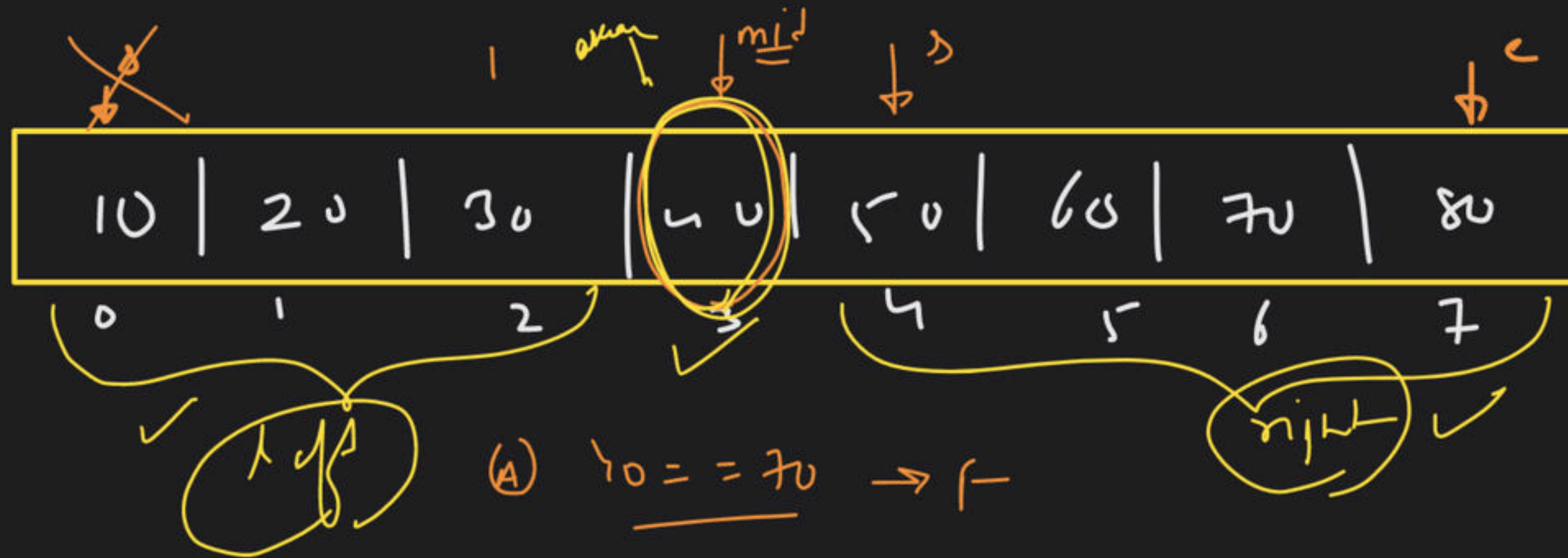
```
    {  
        if (arr[mid] == Key)  
            return mid;
```

```
        if (arr[mid] < Key)  
            s = mid + 1;
```

```
        else  
            e = mid - 1;
```

```
    }  
    return -1;
```

$s = 0$   
 $e = 7$   
 $mid = \left( \frac{s+e}{2} \right) = 3$

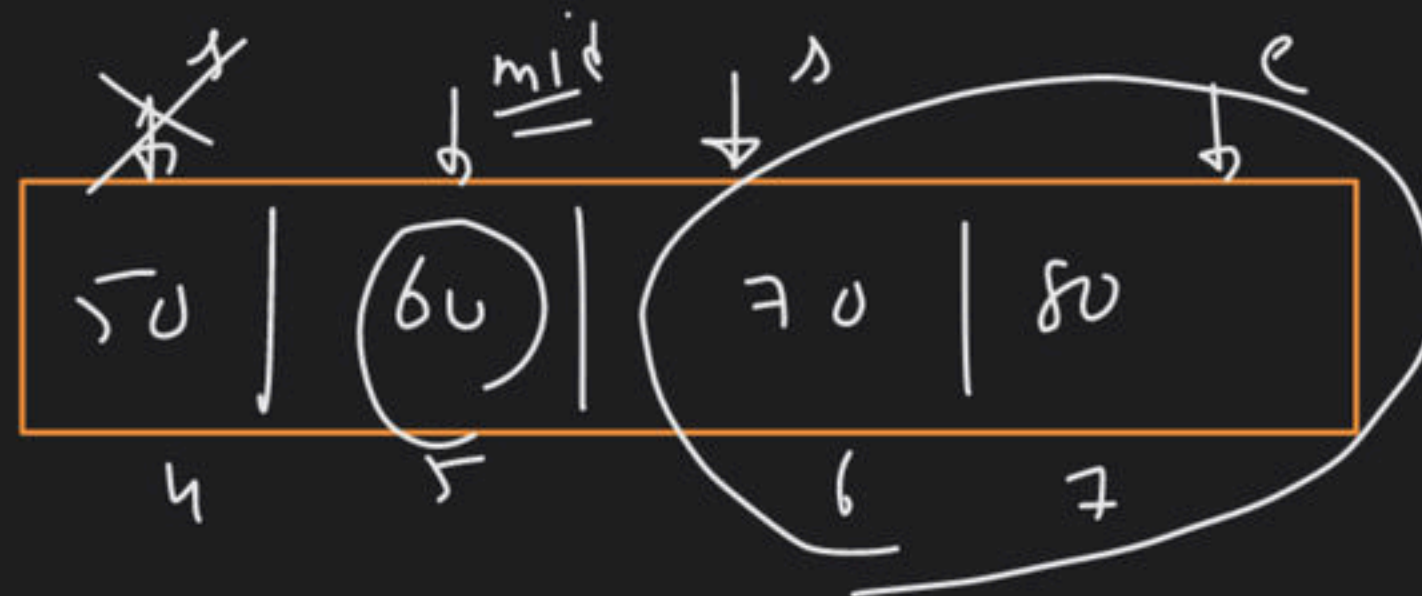


Key = 70

(A)  $40 == 70 \rightarrow f$

(B)  $40 < 70 \rightarrow T \rightarrow \text{Right} \rightarrow s = mid + 1$

$s = 4$   
 $e = 7$   
 $mid = \left( \frac{s+e}{2} \right) = 5$



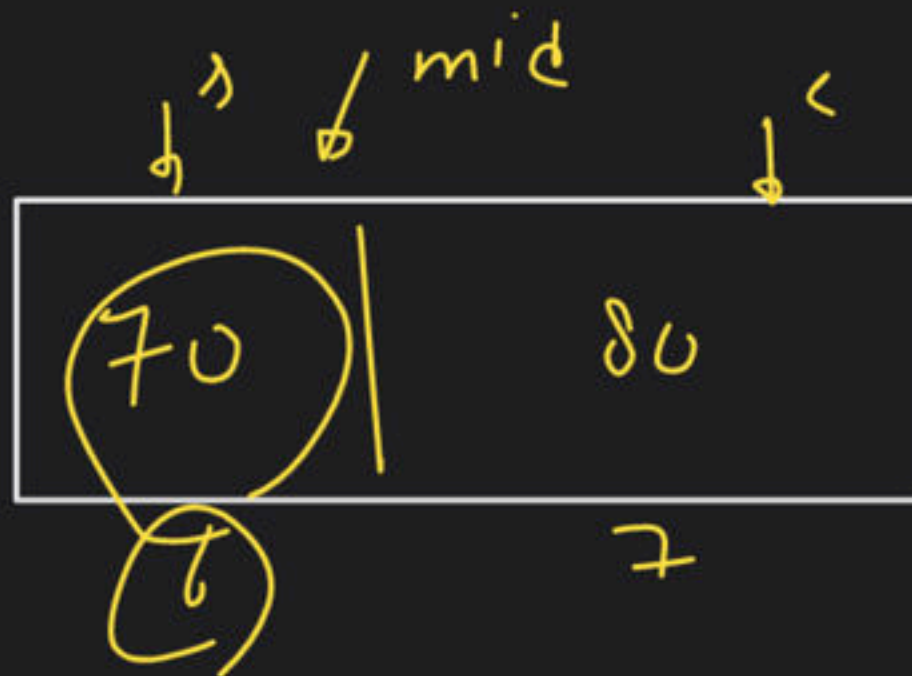
(A)  $60 == 70 \rightarrow f$

(B)  $60 < 70 \rightarrow T \rightarrow \text{Right} \rightarrow s = mid + 1$

$$s = 6$$

$$e = 7$$

$$\text{mid} = \frac{s+e}{2} = (6)$$

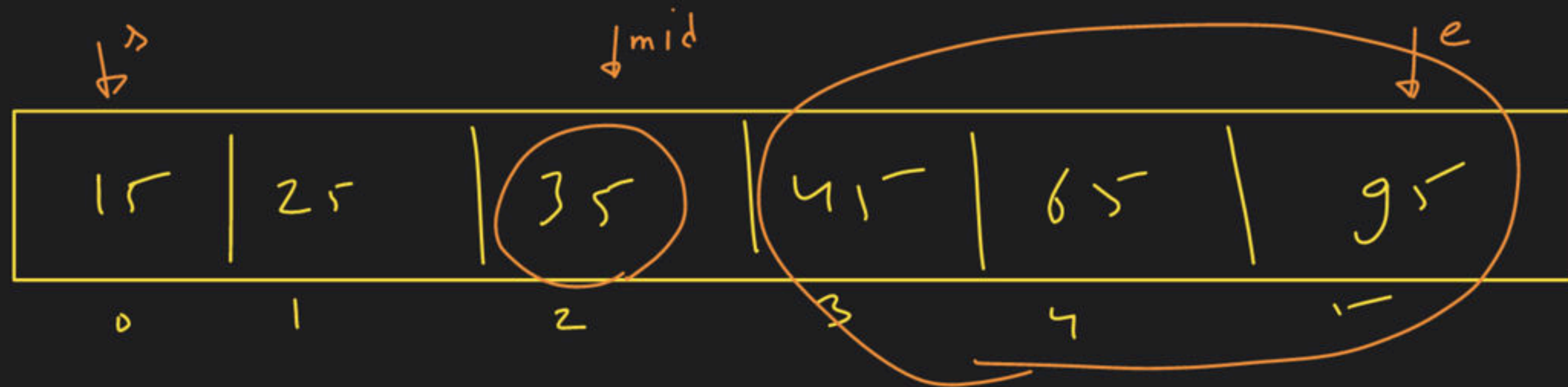


(A)  $70 == 70 \rightarrow \text{True} \rightarrow \text{return mid}$   
return (6)



$s = 0$   
 $e = 5$

$s > e \rightarrow F$   
 $mid = 2$

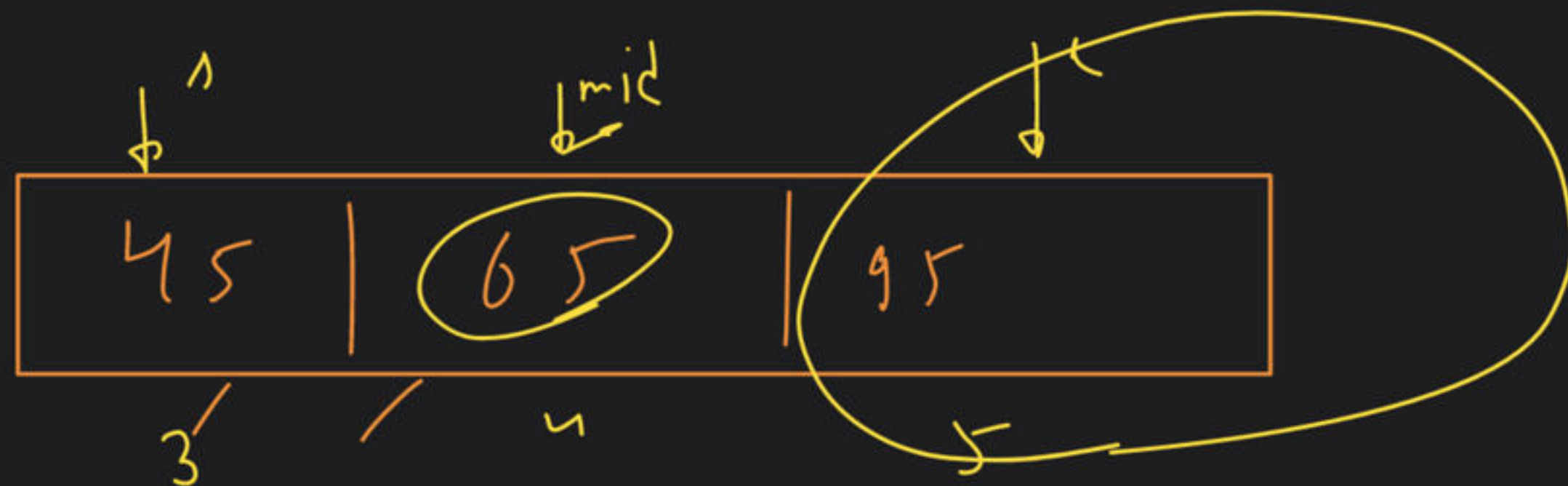


(A)  $35 == 95 \rightarrow F$

(B)  $35 < 95 \rightarrow True \rightarrow \text{right side}$

$s = 3$   
 $e = 5$   
 $mid = 4$

$s > e \rightarrow F$

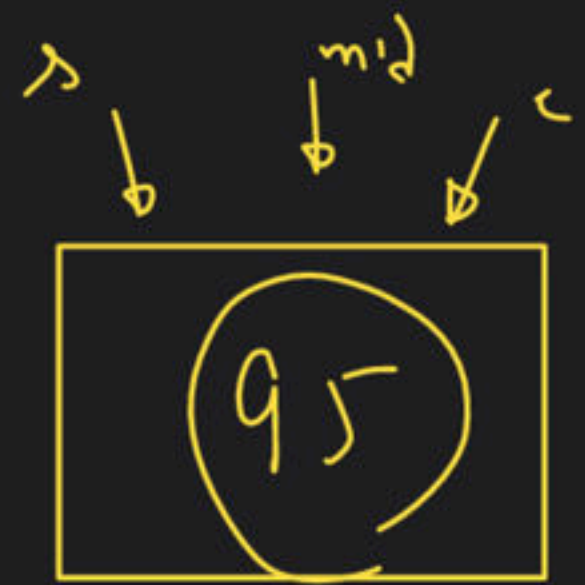


(A)  $65 == 95 \rightarrow F$

(B)  $65 < 95 \rightarrow True \rightarrow \text{right}$

s = 5  
e = 5  
mid = 5

$s > e \rightarrow f$



5

(A)

$95 == 95$

True

return 5



~~PB~~ → 6MB → ~~90~~  
~~PBR~~ → ~~250MB~~ → 90ms

PBV → 250MB → 90ms  
PBR → 6MB → 90ms

1000

1000000

1000 len  
↳ 1000 freq

linear

4 byte

16 byte

vector<int> & v



6 elements

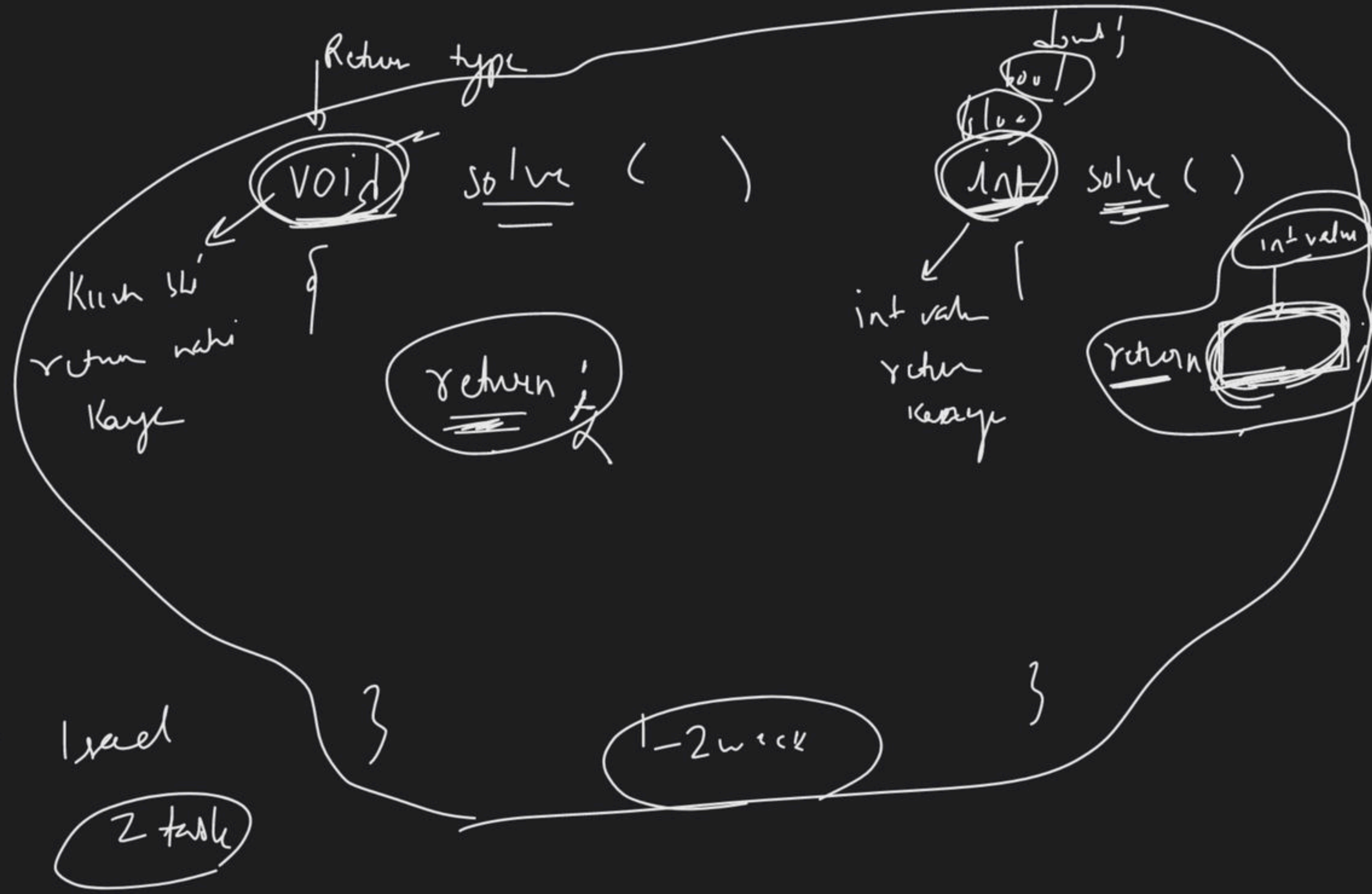


10

00  
00

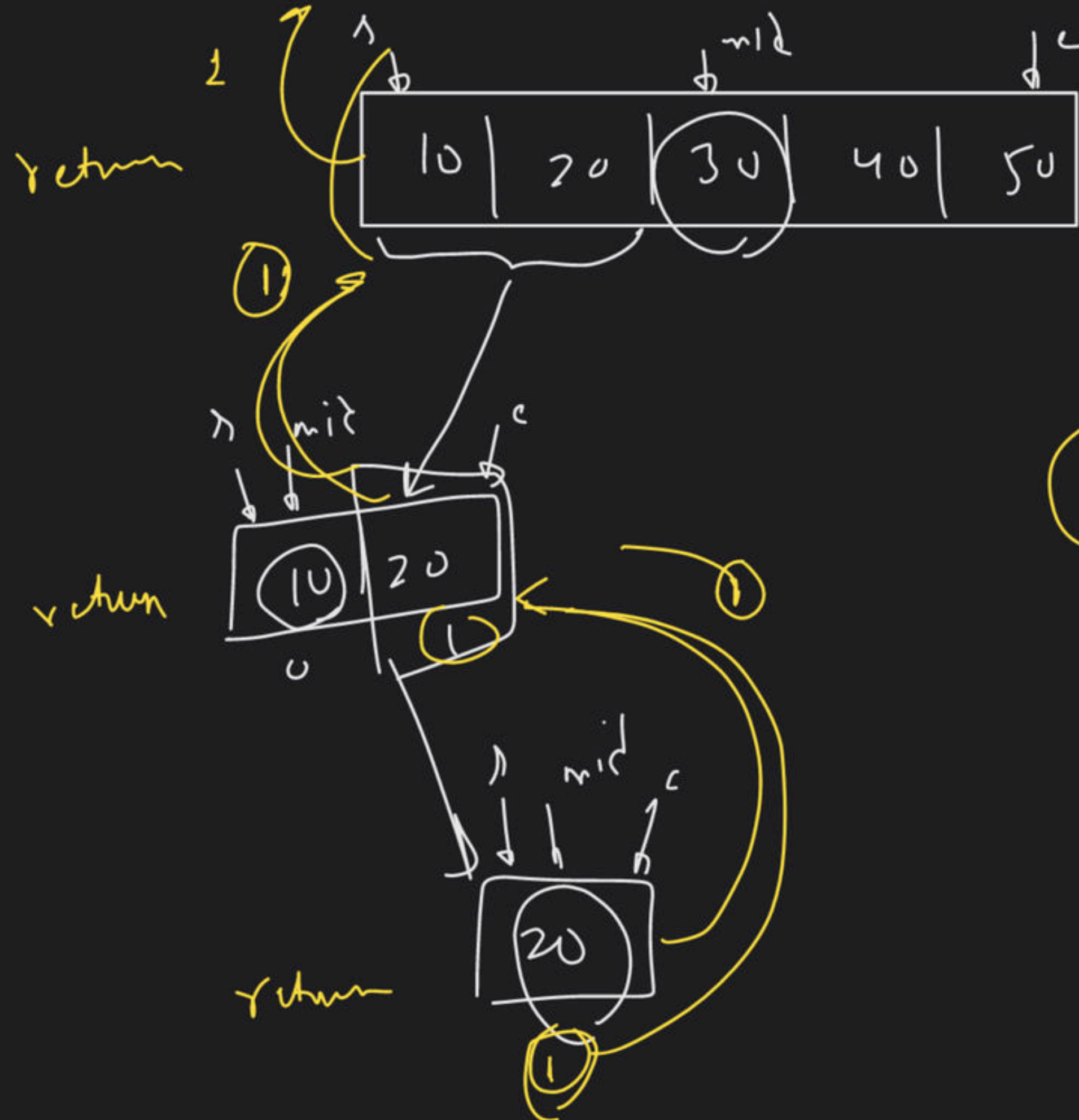






2 → 20





Key = ~~10~~  
20

int Bs ( )  
{  
  
}

③

# Subsequences of a String

famous pattern

include - exclude  $\rightarrow n=3 \rightarrow 2^3 \rightarrow 8$

i/p  $\rightarrow$  "a b c"

o/p  $\rightarrow$  print all subsequence

a b c

✓ ✓ ✓  $\rightarrow$  a b c

✓ ✓ x  $\rightarrow$  a b

✓ x x  $\rightarrow$  a

x x x  $\rightarrow$  -

x ✓ x  $\rightarrow$  b

x x ✓  $\rightarrow$  c

x ✓ ✓  $\rightarrow$  b c

✓ x ✓  $\rightarrow$  a c

power set

{-}, {a} {b} {c}

{a, b} {a, c} {b, c}

{a, b, c}

i/p  $\rightarrow$

"xy"  $\rightarrow n=2$

$$2^n = 2^2 = 4$$

xy

✓ ✓  $\rightarrow$  xy

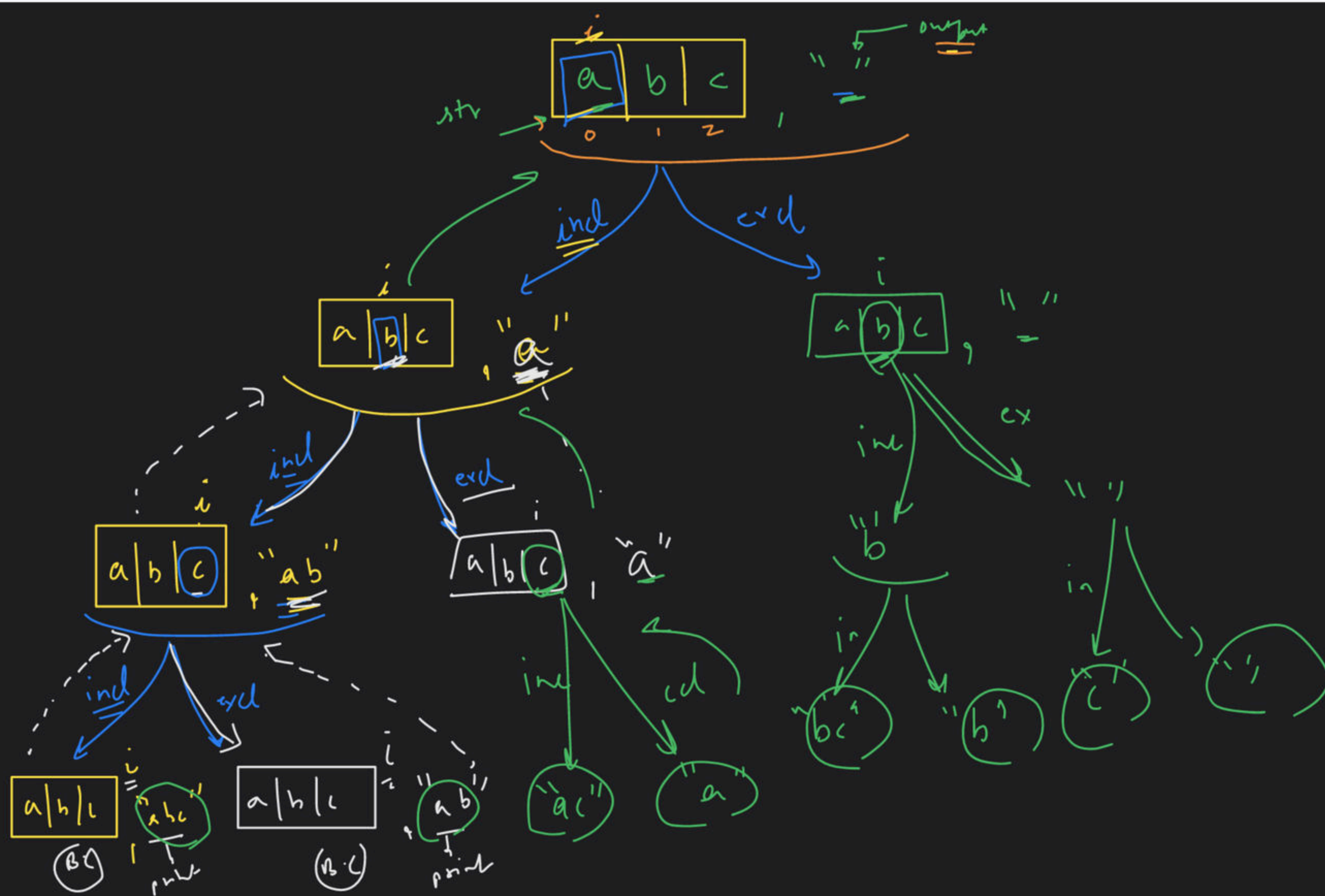
✓ x  $\rightarrow$  x

x ✓  $\rightarrow$  y

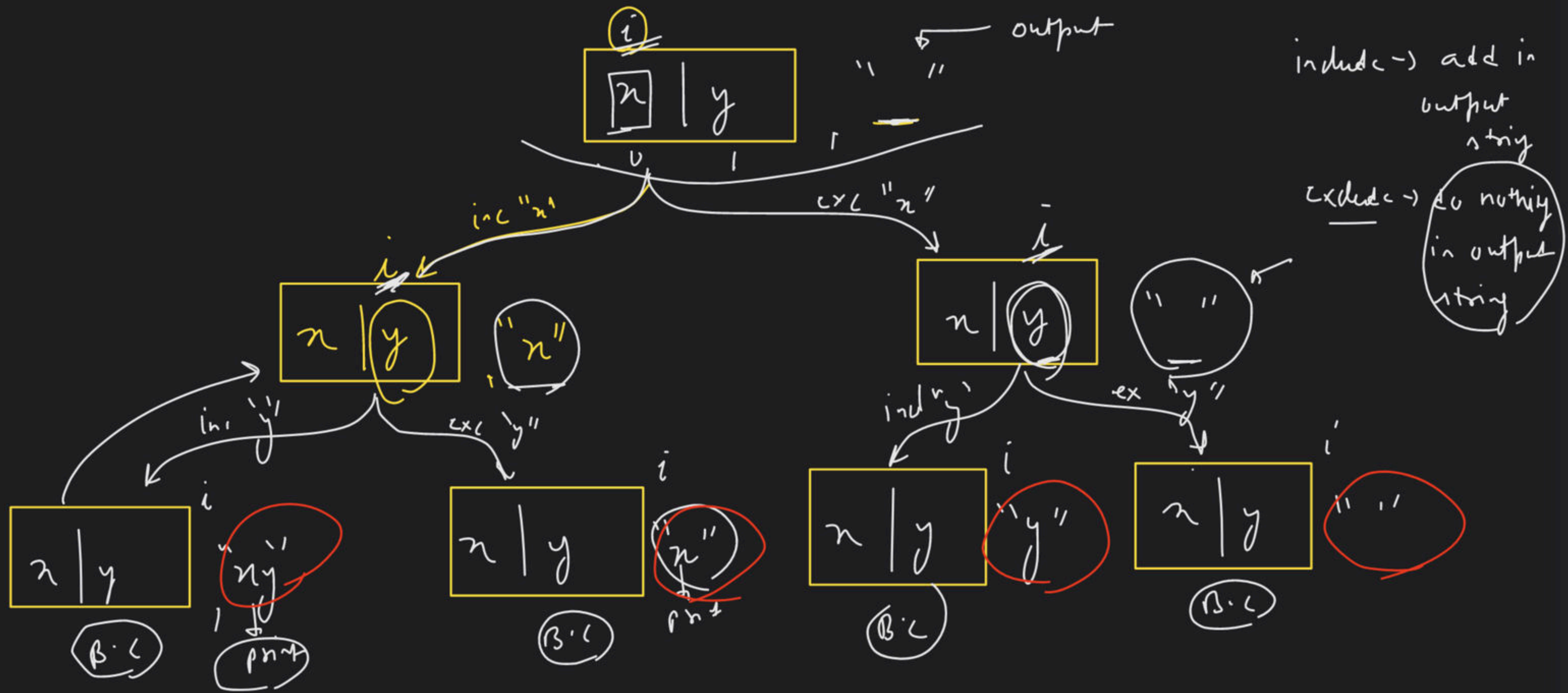
x x  $\rightarrow$  -









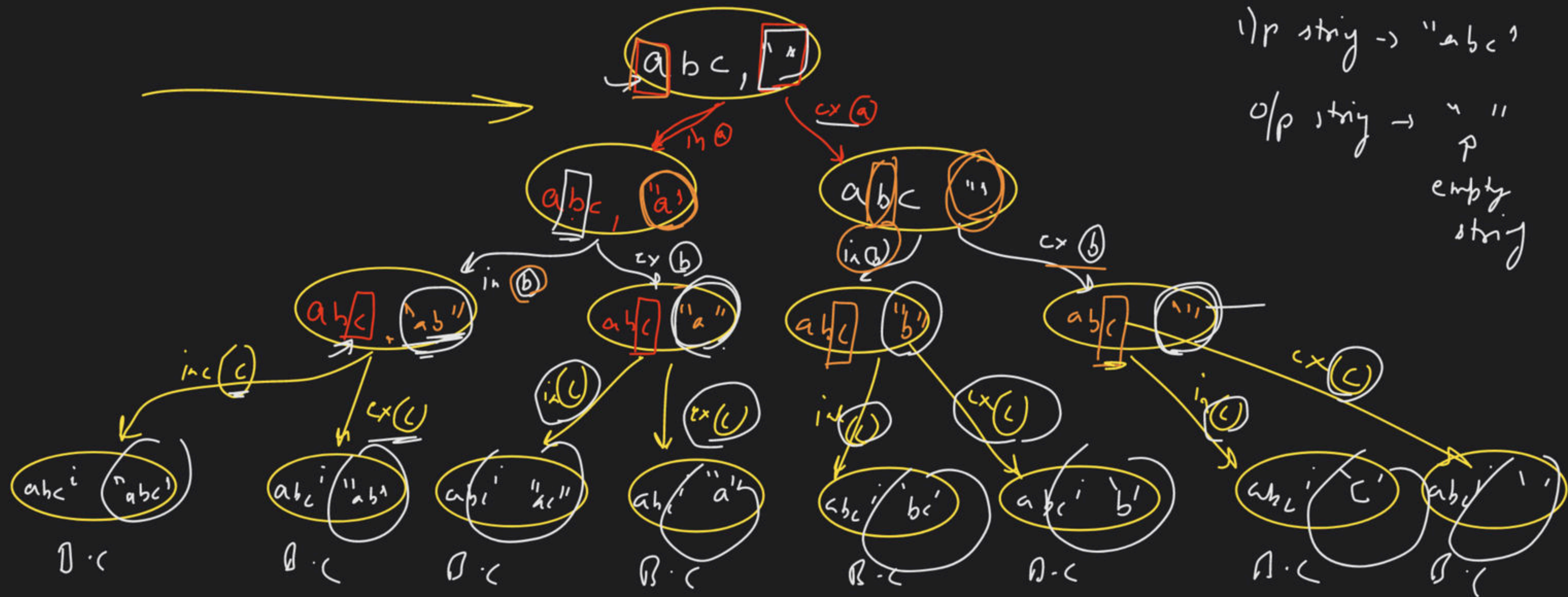


$$"" + y = (y)$$

$$"" + n \rightarrow (n)$$

$$"n" + "y" \rightarrow ny$$





1/p string  $\rightarrow$  "abc"

o/p string  $\rightarrow$  "p"  
empty string

" " + a  $\rightarrow$  a

" " a

"a" + "b"

"a"  $cx(c)$  "b"

"a" + "b"

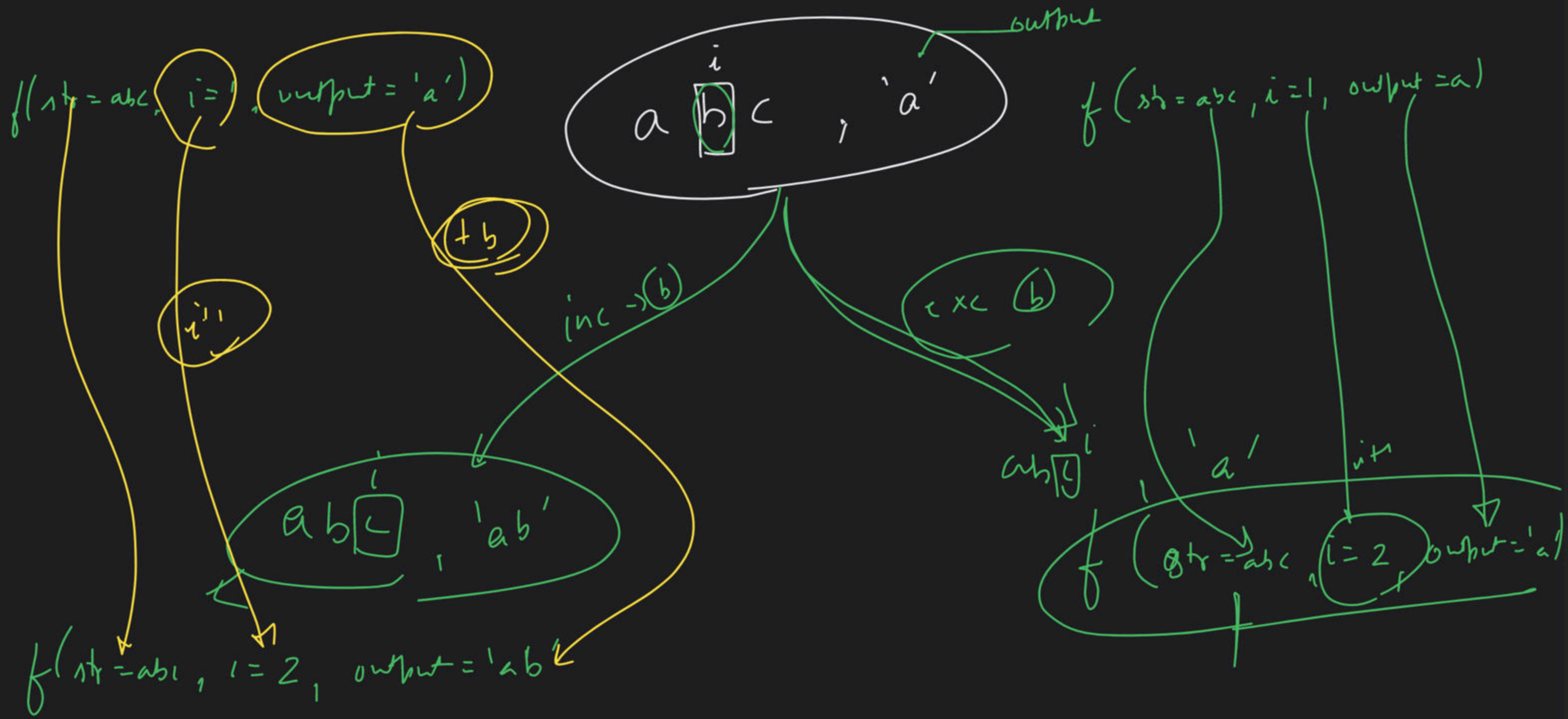
include → concat  
" " + "n" → "n"

---

evaluate → as it is

~~///~~ "x" → cx "n"  
as it is







Subsequence



Bitmasking











