infix → 2 + 6 * 4 / 8 - 3

2 + 24 / 8 - 3

2 + 3 - 3

5 - 3

2

Left

*/

+-

( )* 2

pc → infix a + b

pre

post

+ a b

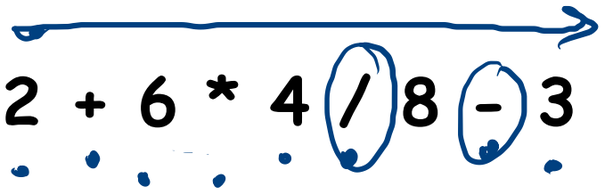
a b +

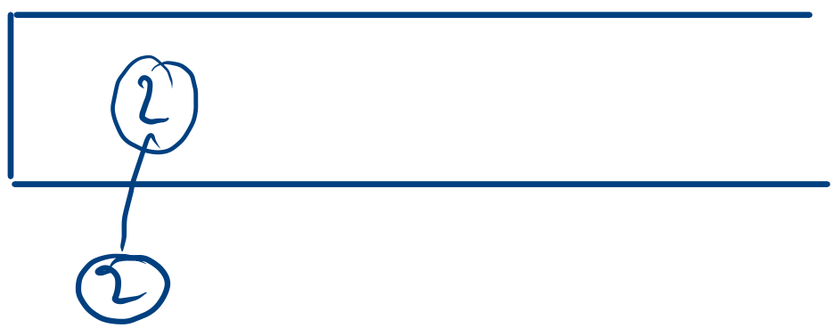
a+b

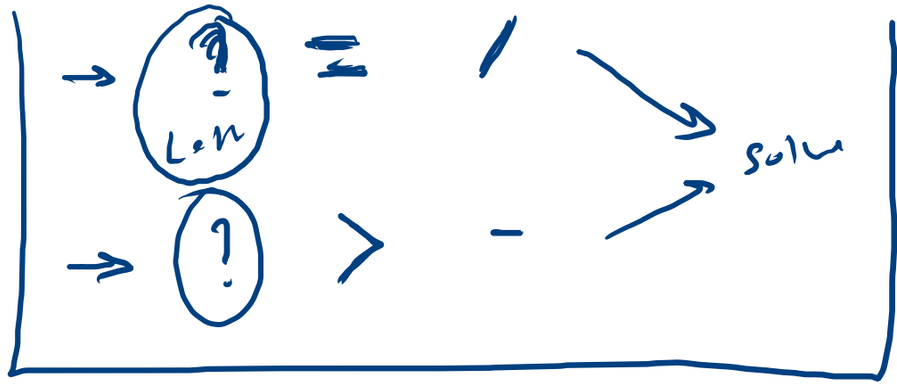
postfix

*/

+-







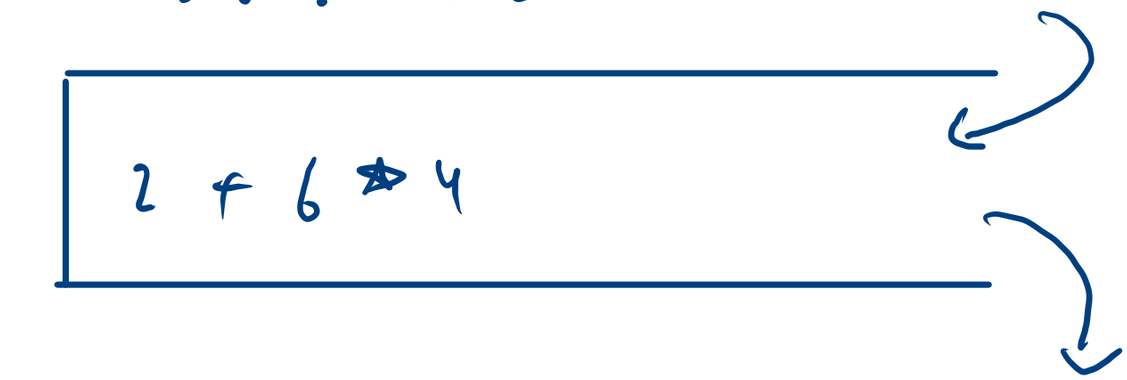
2 + 6*4 / 8 - 3

2 + 24 / 8 - 3

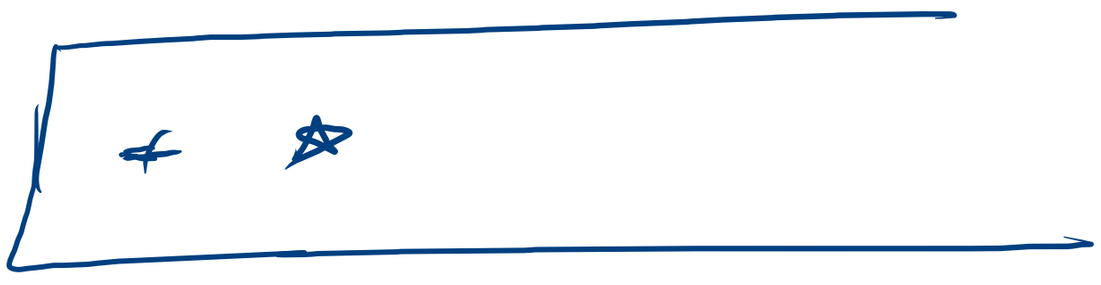
2 + 3 - 3

5

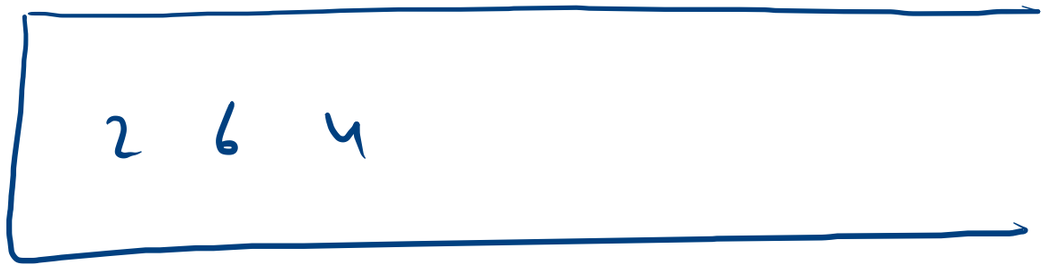
2 + 6 * 4 / 8 - 3



operator



number



! ≥ curr → solve

* / → 2

+ - → 1

1

2 + 6 * 4 / 8 - 3

2

V1 + V2

operato

5 - 3 = 2

op = -

? ≥ (or → solve

* ≥ /

/ ≥ -

+ ≥ -

* / → 2

+ - → 1

2

number

V2 = 3

V1 = 5

? ≥ curr

* / → 2

+ − → 1

2 + 6 * 4 / (8 - 3)

2 + 6 * 4 / (8 - 3)

2 + 6 * 4 / 5

2 + 24 / 5

2 + 24 / (8-3)

2 + 24 / 5

2 + 4

6

while(op.peek != '(')
    solve

op.pop

) → loop

2 + 6 * 4 / (8 - 3 * 1) + 2

2 + 24 / 5

v1 - v2

? ≥ curr

+ ≥ *

* ≥ /

+ ≥ /

(≥ -

/ → 2

+ - → 1

(→ 0

-> = *

/ >= +

number → push value

operator → check op.peek solve

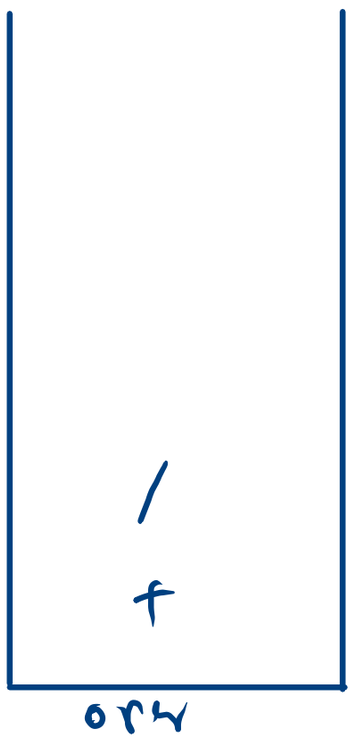
add curr

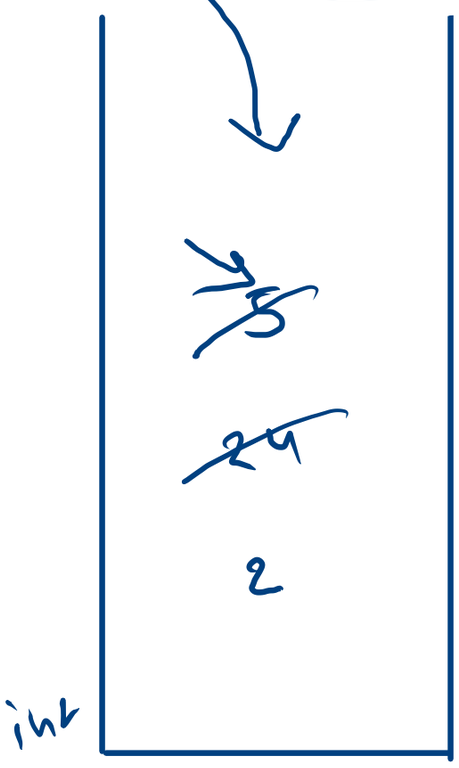
(→ add op

) → solve until (

(→ pop

solve





a + b * c - d / e + f - g

Important

a + n

exp

(a - b * c)

? < -

exp (a - b / y)

O(n)

( )

n/2

exp-1 / 2

a - b * c

? < -

a + b * c

space complexity

exp 100

((((

so so

((((((((

a * b + c

max (3)

n + c

a + b * c + d

a + b * c

a - b - c

n - c

constant

prefix

a*(b-c+d)/e

a* -bc +d /e

a* +-bc d /e

* a +-bc d /e

/ * a +-b c d e

v1 op v2

op v1 v2
pre

v1 v2 op
post

priority →

~~priority()~~

infix → a+b

prefix → +ab

post → ab+

a*(b-c+d)/e

a-b

-bc

v1 op v2 → op v1 v2

v1 op v2 → v1 v2 op

? ≥ curr

- ≥ +

* ≥ /

int b = ?

int c = ?

int ans = b - c → -bc

op = /

/ * a + - b c d e

v2 = e

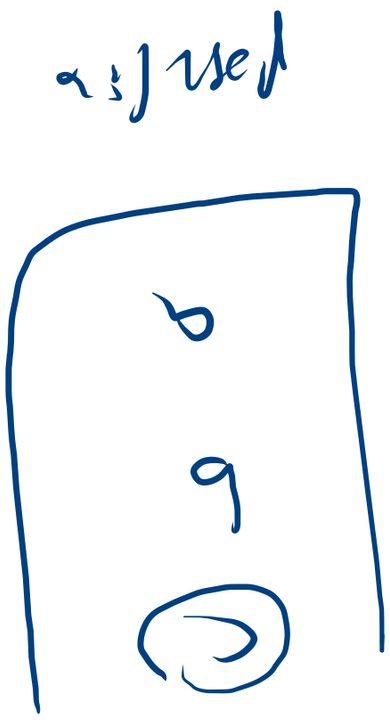
v1 * a + - b c d

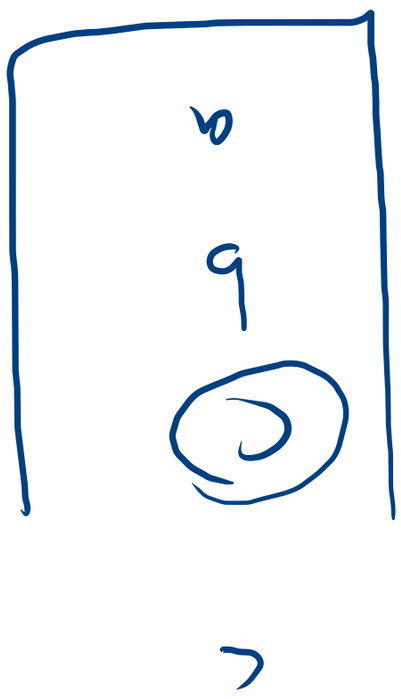
character

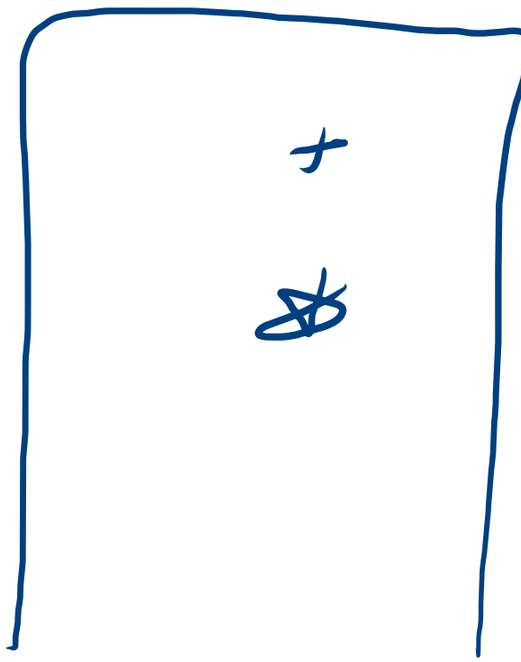
Operator

Stack

Postfix


a
b
c


a
b
c


+
*


a + b * c

264*8/+3-

ab*

2 24 8 / +3 -

a b /

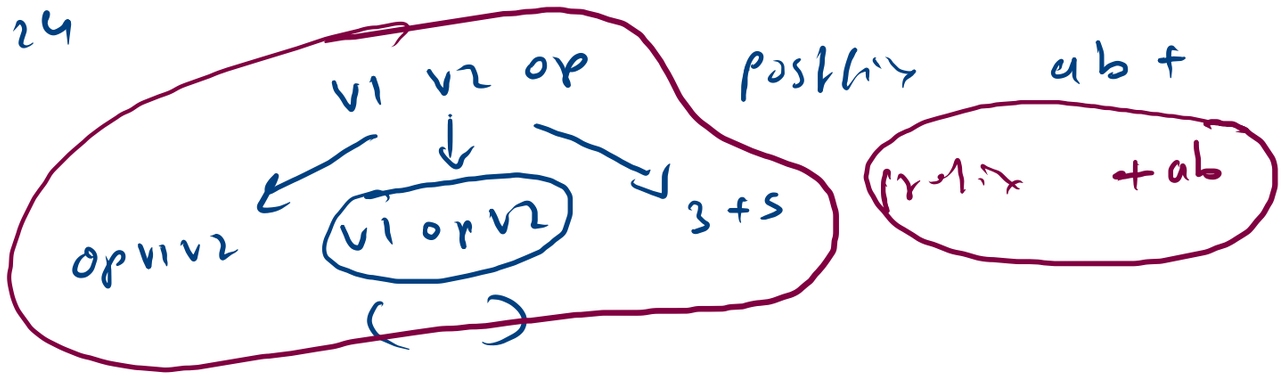
2 3 +3 -

a b +

5 3 -

(2)

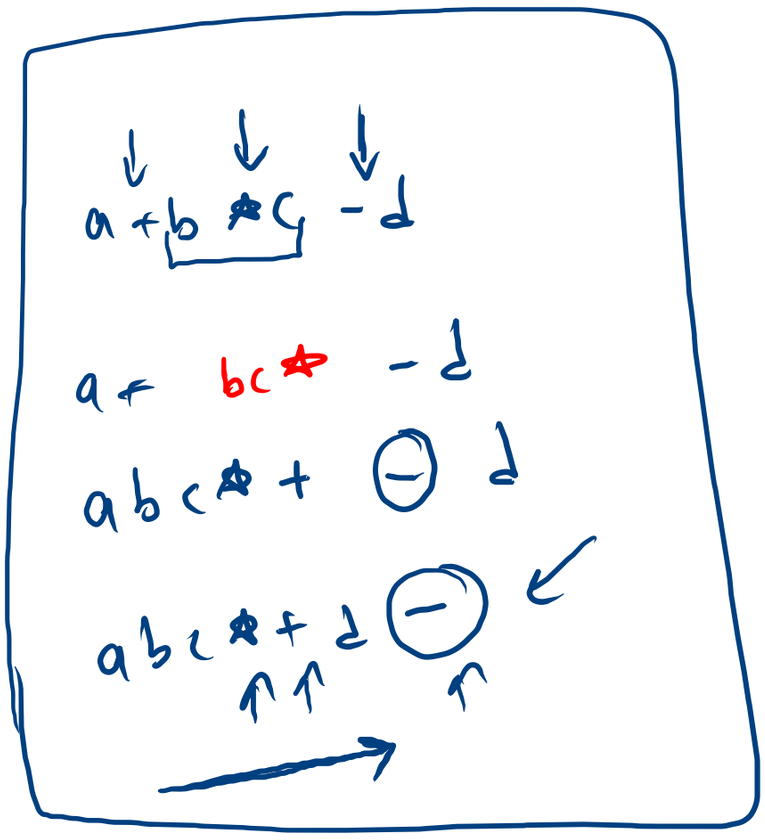
6×4 = 24

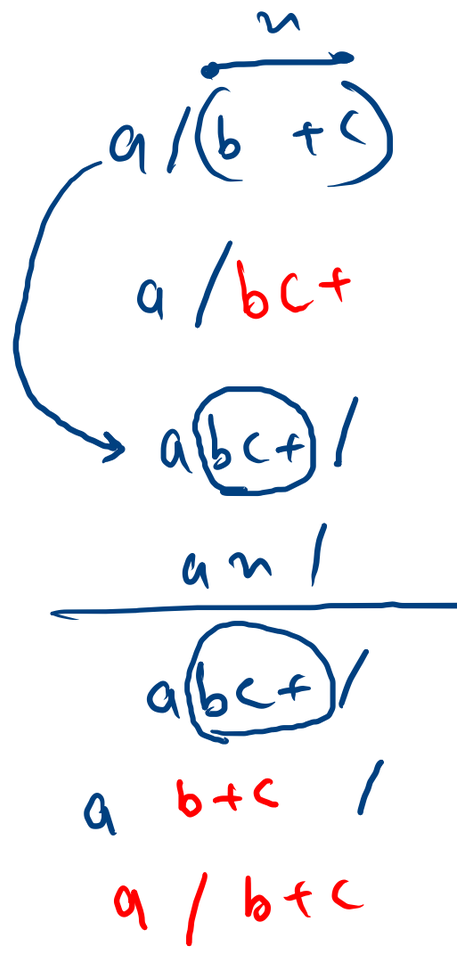


a+b-c

ab+ c

ab+c -





☆ prev... [prefix eval conv] H.W.

★ Direction v1 v2 op

264*8/+3-

2 24 8/ +3-

2 3 + 3-

a b +

5 3-

24

264*8/+3-

2 4

V1 V2

5 - 3

V2

V1

op

V1 op V2

V1 V2 op

op = *

V2 = 4

V1 = 6

6*4 = 24

postfix

V2 = 8

V1 = 24

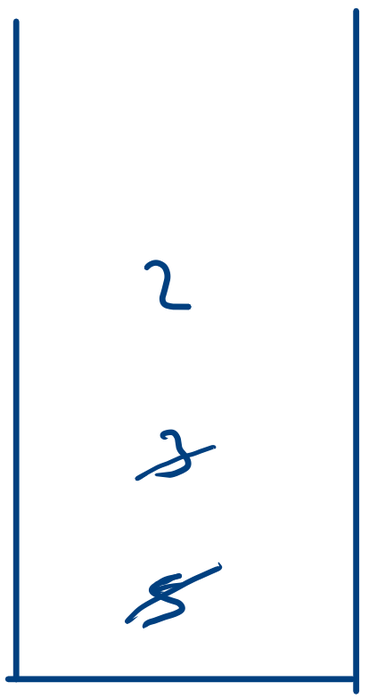
op = /

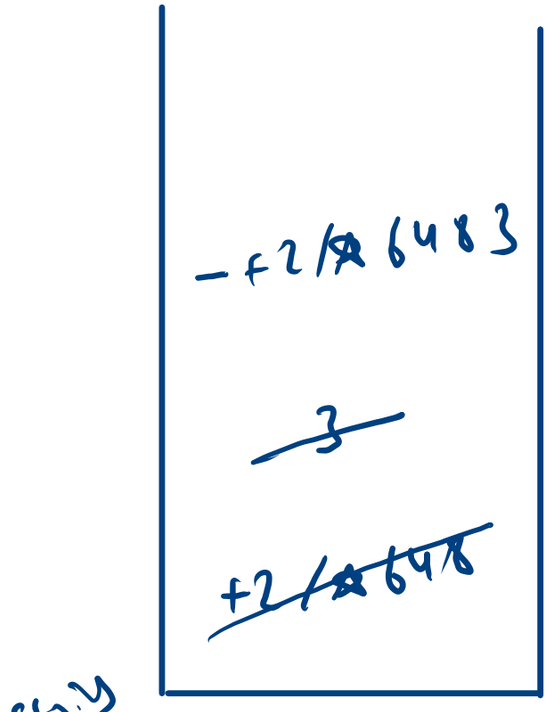
V2 = 3

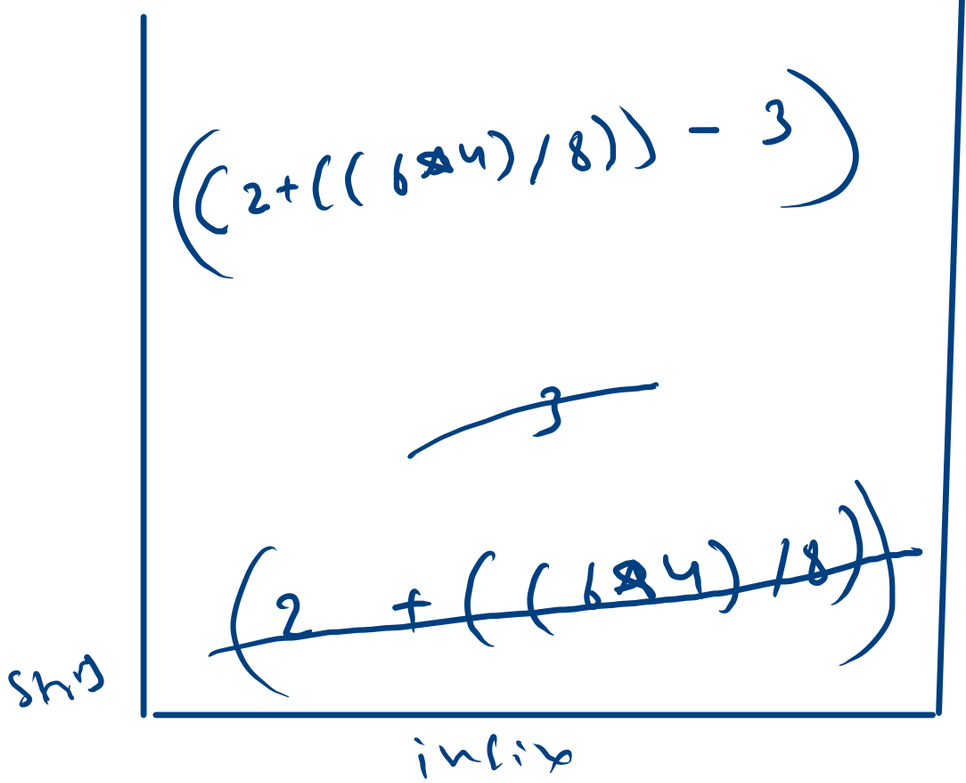
V1 = 2

264*8/+3-

6x4


2
3
5
number
value


-+2/*6483
3
3
+2/*648
stry
prefix


((2+((6*4)/8))-3)
3
(2+((6*4)/8))
stry
infix

$h = y$

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

$mat[x][y] = 1$

x knows y

$mat[x][y] = 0$

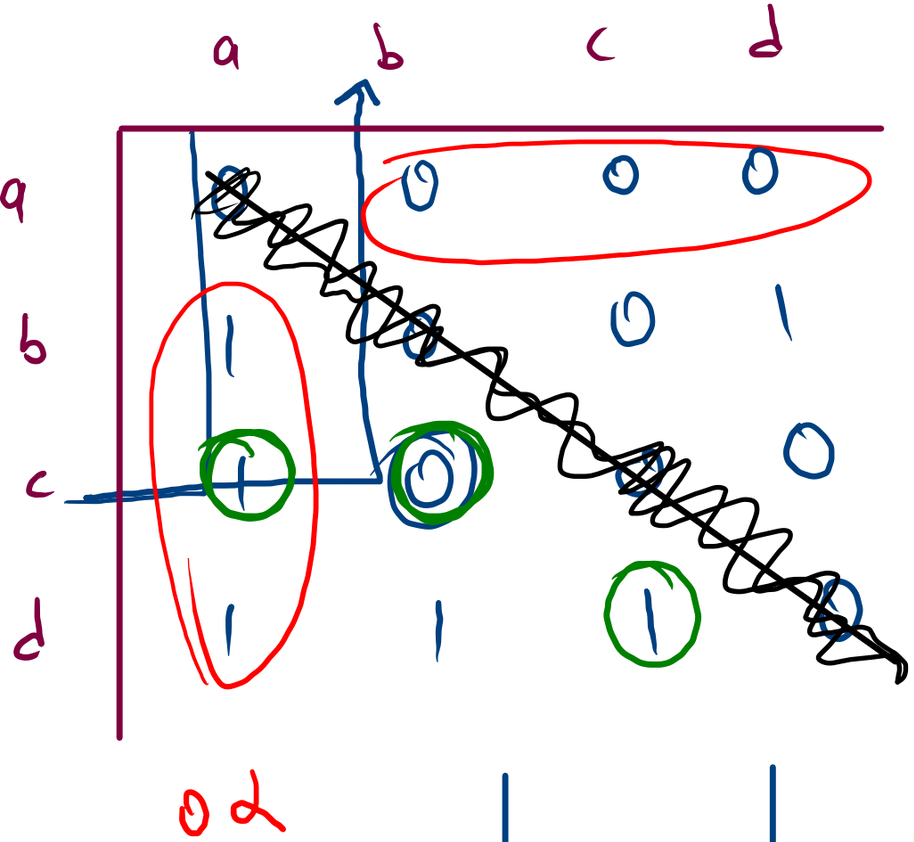
x dont know y

(b) a < b

- Celebrity know no one
- everyone know celebrity

a $\not\rightarrow$ b

b $\not\rightarrow$ a



1 α

0 α

a

$ar[n][y] = 1$ ✗ ✓

n y

$n \rightarrow y$

$ar[n][y] = 0$ α ✓

y n

n does not know y

$ar[d][c] = 1$

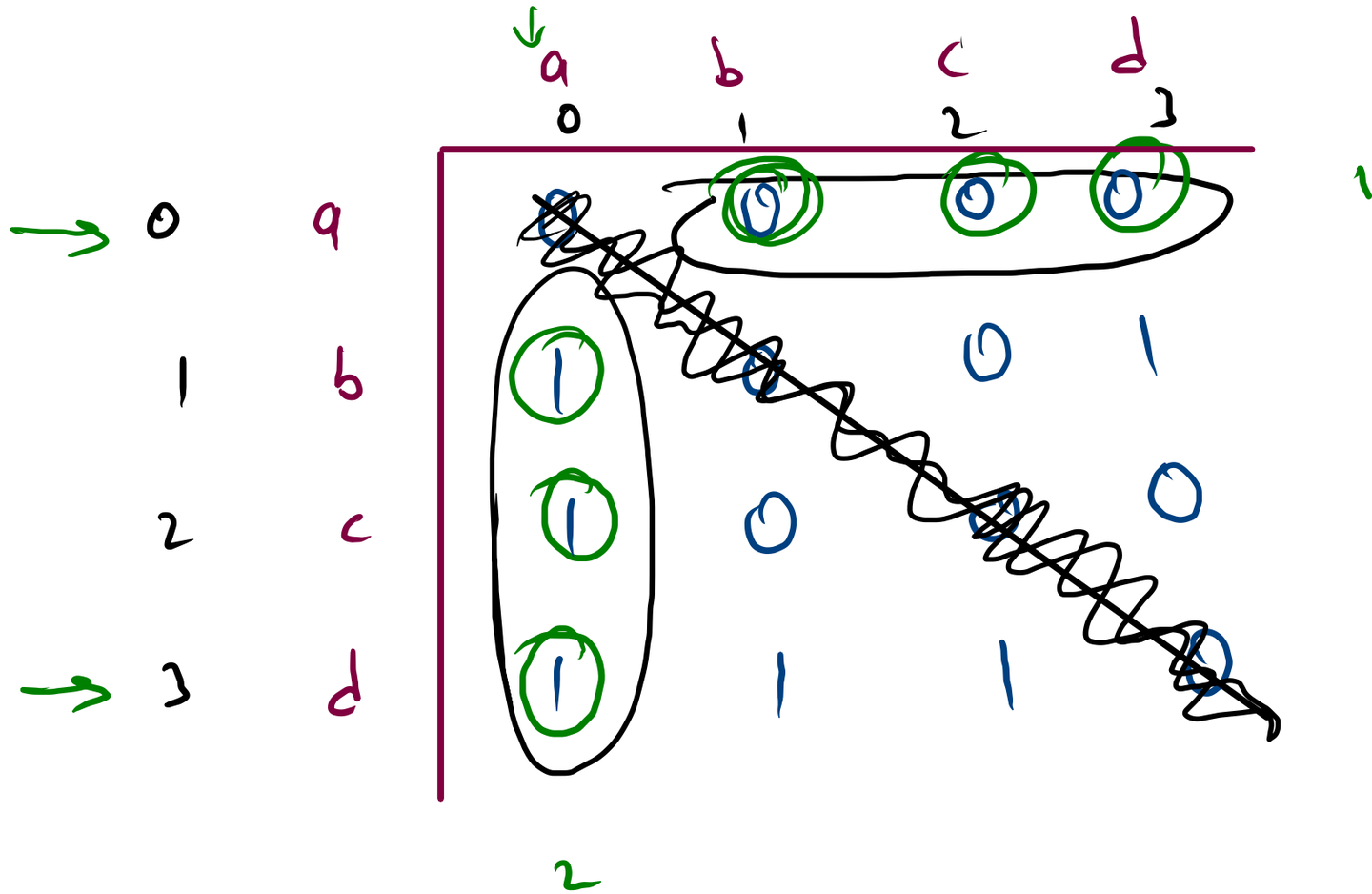
$ar[c][b] = 0$

$ar[a][q] = 1$

a

a = 0

```java
boolean cel = true;
for(int i=0;i<arr.length;i++){
    if(i==a)continue;
    if(arr[a][i] == 1 || arr[i][a]==0){
        cel = false;
        break;
    }
}
if(cel){
    System.out.println(a);
}else{
    System.out.println("none");
}
```



i = 0 ~ < 3

arr