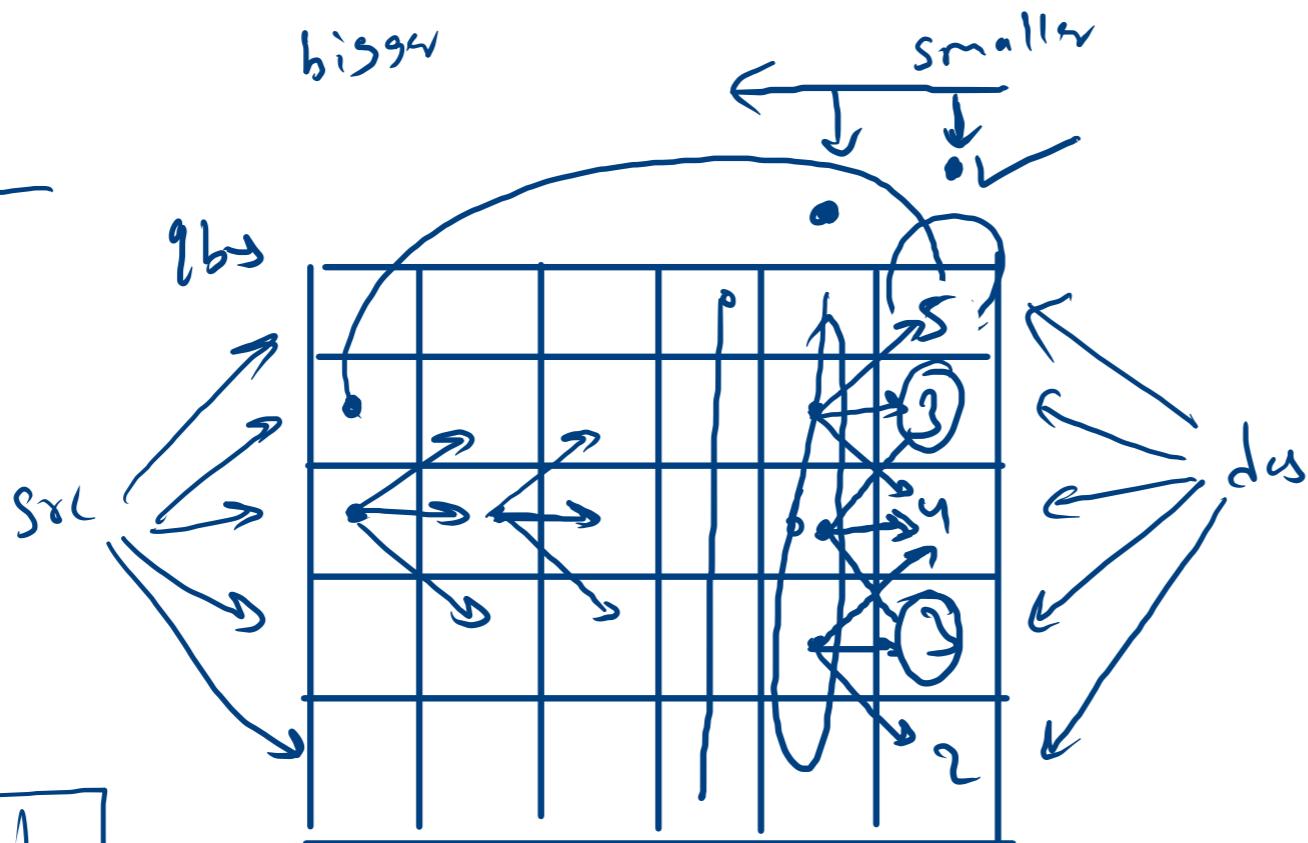
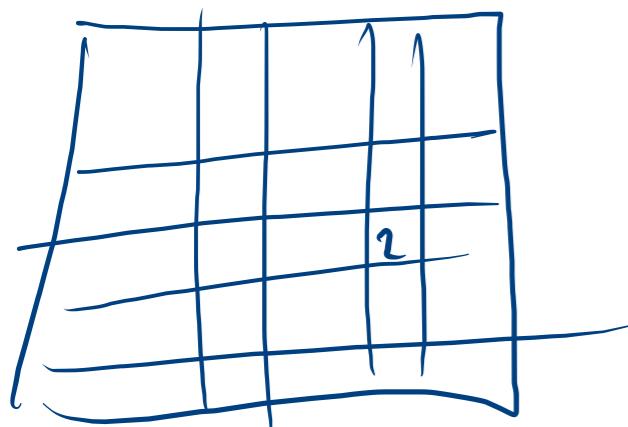


$s_r := drss$
 $s_c := dc$



```
for (int c = m-1; (c >= 0); c--)  

    for (int r = 0; r < n; r++)
```

3

0/1

5

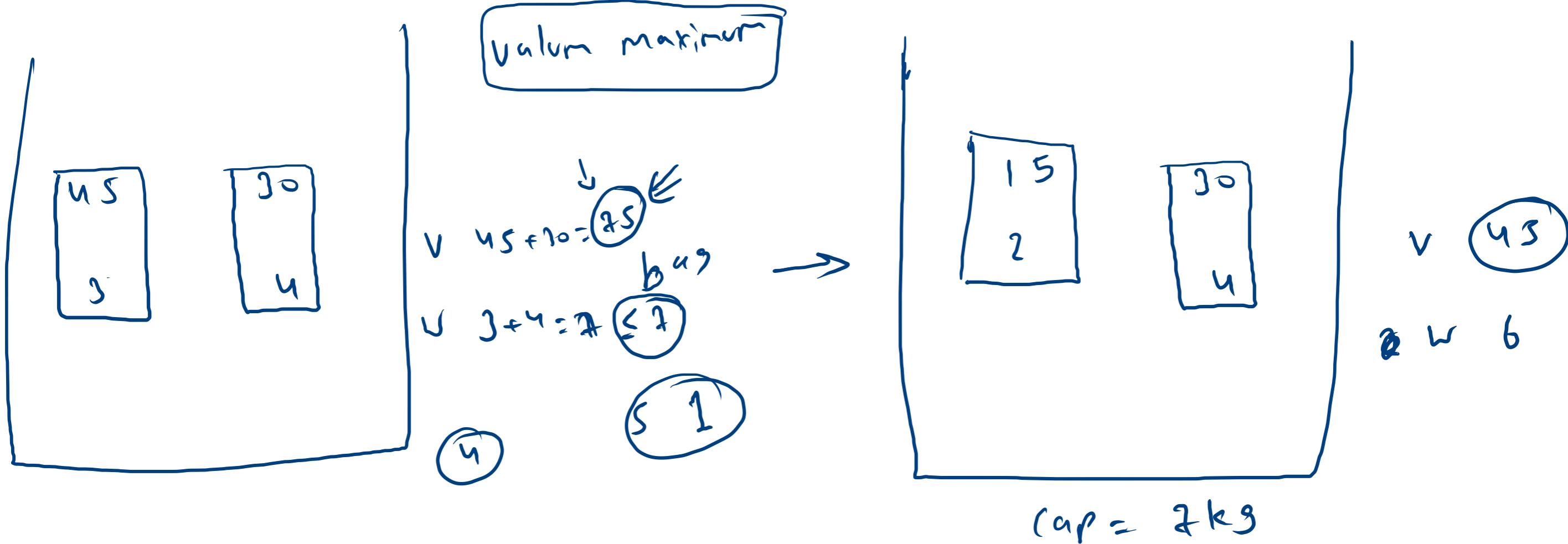
v 15 14 10 45 30

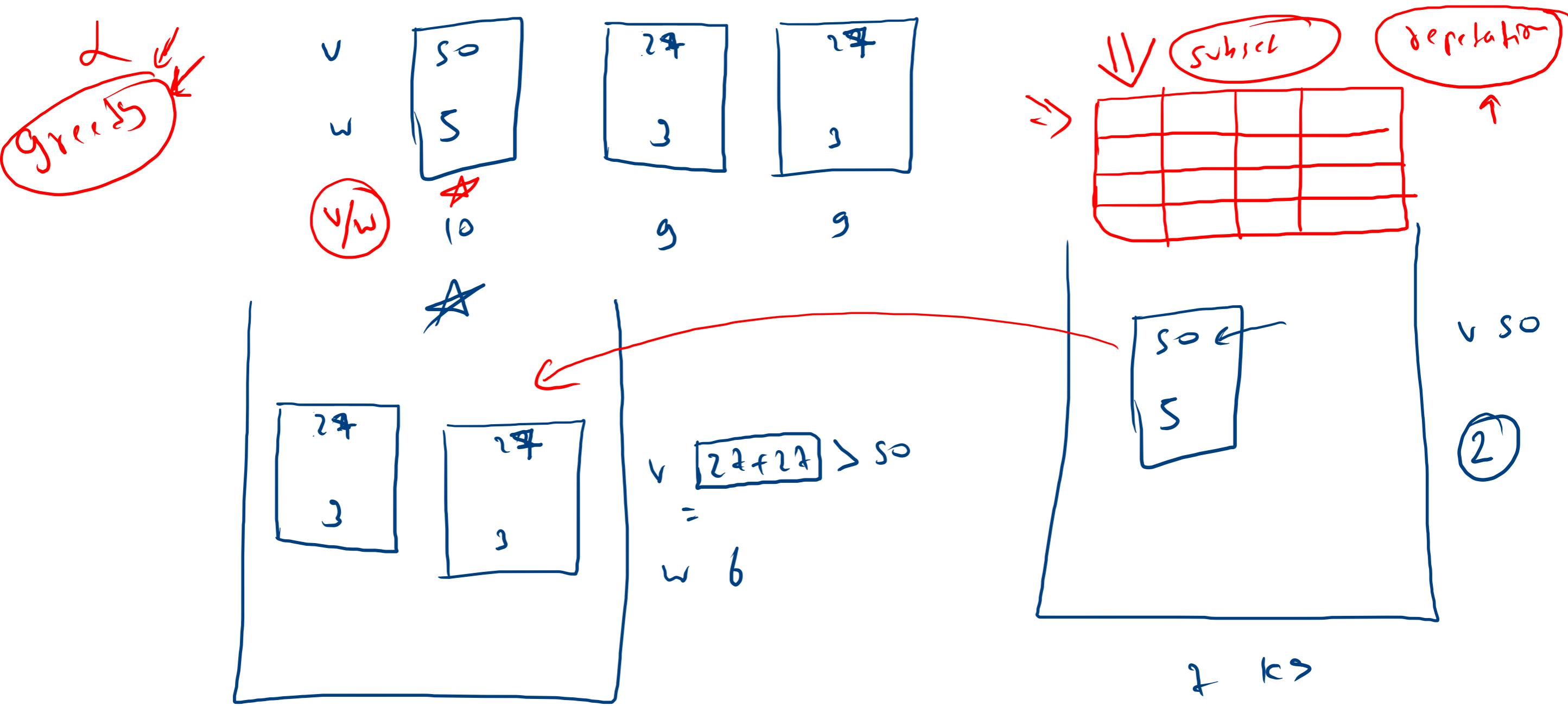
w 2 5 1 3 4

Cap 7

v	15 2	14 5	10 1	u5 3	30 4
---	---------	---------	---------	---------	---------

→ 75

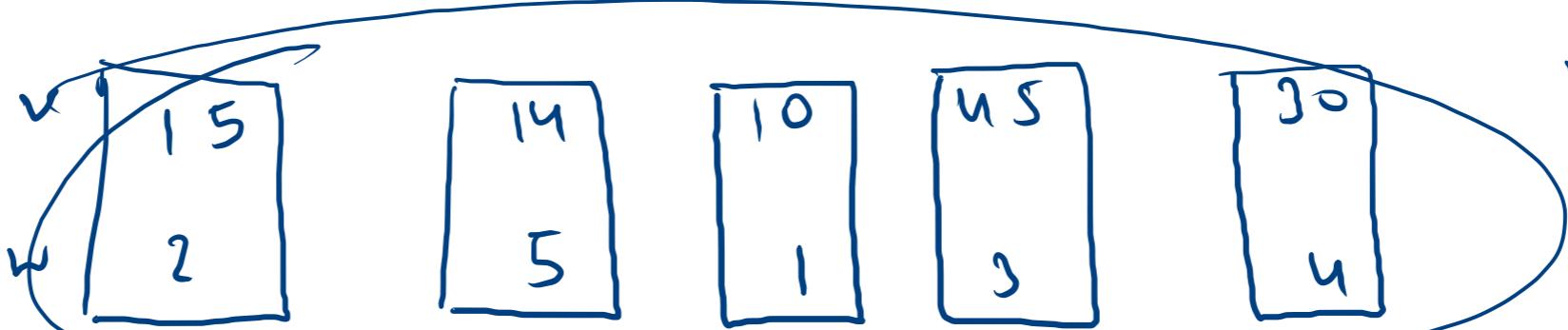




0/1

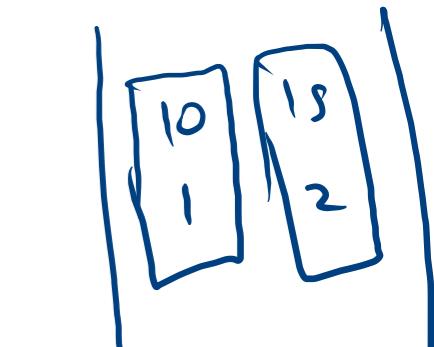
5

V 15 14 10 45 30
W 2 5 1 3 4

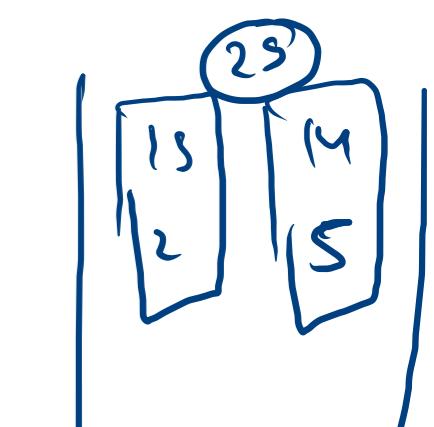


P 7

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
15	0	0	15	15	15	15	15	15
14	0	0	15	15	15	15	15	15
10	0	10 - 0	10 - 15	25	15	25	25	25
US	0	10	15	45	25	45+10, 25	45+25, 25	45+25, 25
30	0	10	15	45	55	60	70	20
45	0	10	15	45	55	60	70	75



25 7 cap



15 7 kg
10 45
14 30
5 25

```

import java.util.*;
Mute Start Video Security Participants
Your

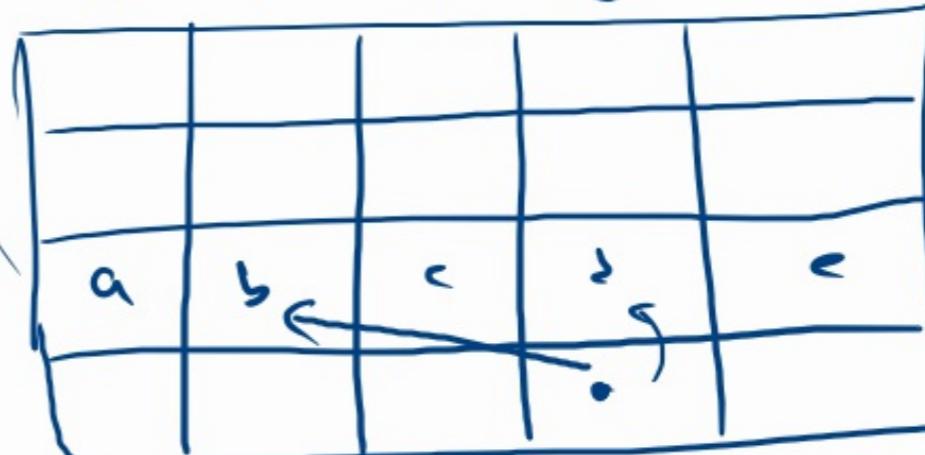
public class Main {
    public static void main(String[] args) throws Exception {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int v[] = new int[n];
        for(int i=0;i<n;i++) v[i] = scn.nextInt();
        int w[] = new int[n];
        for(int i=0;i<n;i++) w[i] = scn.nextInt();

        int cap = scn.nextInt();

        // code
        int qb[][] = new int[n+1][cap+1];
        for(int i=1;i<=n;i++){
            int value = v[i-1];
            int weight = w[i-1];
            for(int c=1;c<=cap;c++){
                // exclude
                int max = qb[i-1][c];

                // include
                if(weight <= c){
                    int rewe = c-weight;
                    max = Math.max(max, value+qb[i-1][rewe]);
                }
                qb[i][c] = max;
            }
        }
        System.out.println(qb[n][cap]);
    }
}

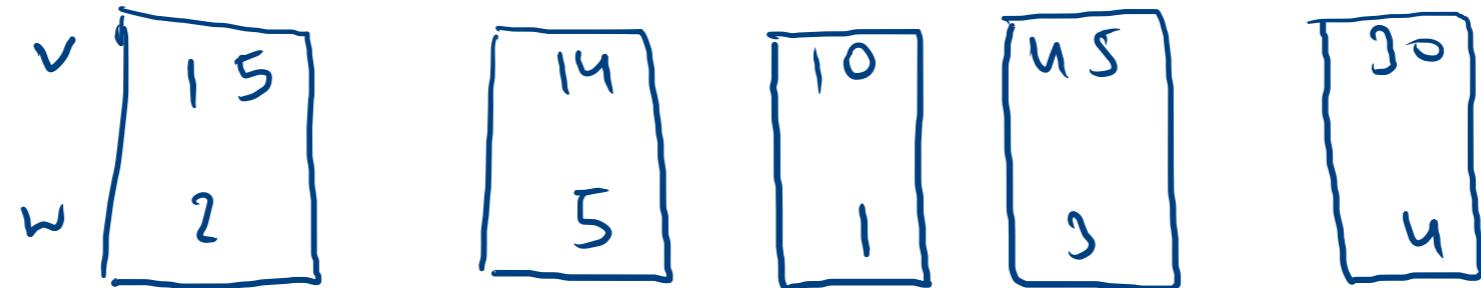
```



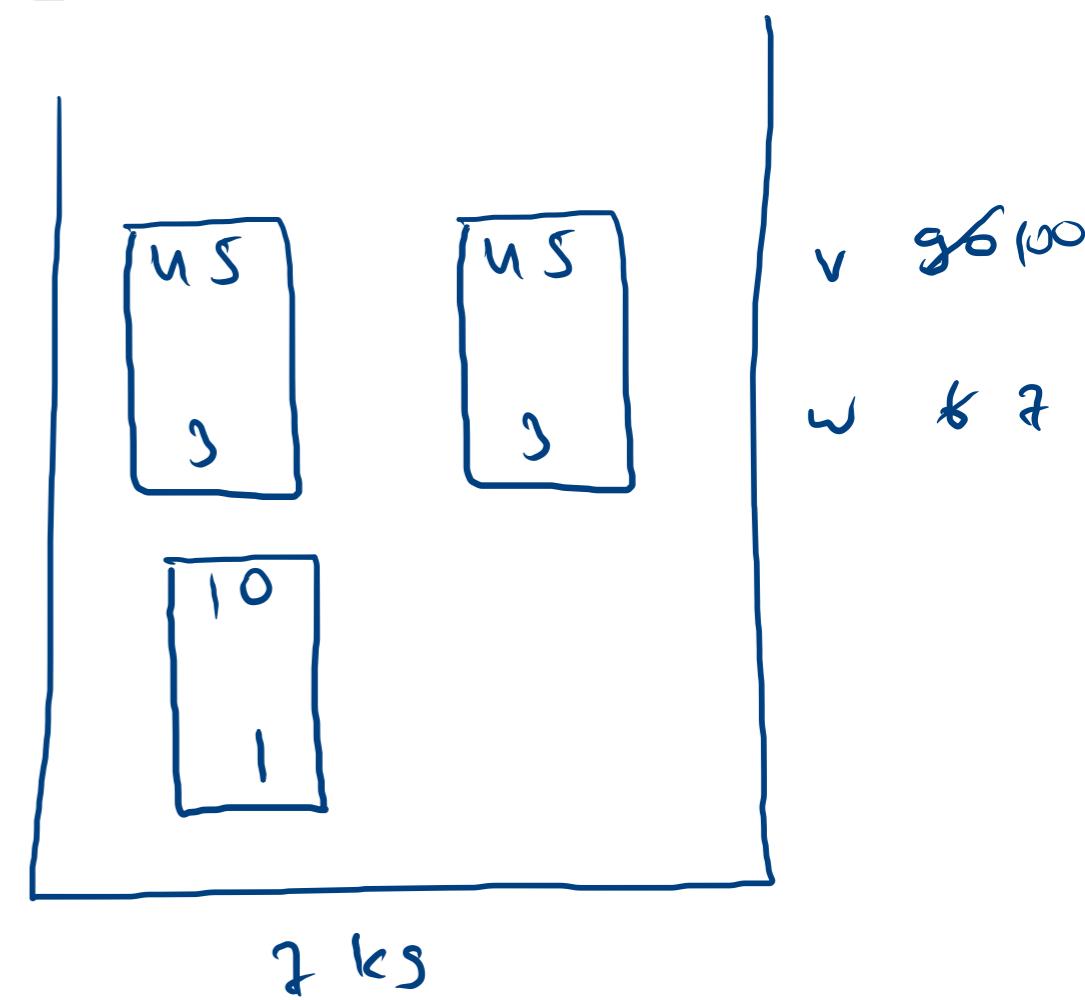
$$\max \left(\left(d, \right) \left(10 + b \right) \right)$$

5
15 14 10 45 30
2 5 1 3 4

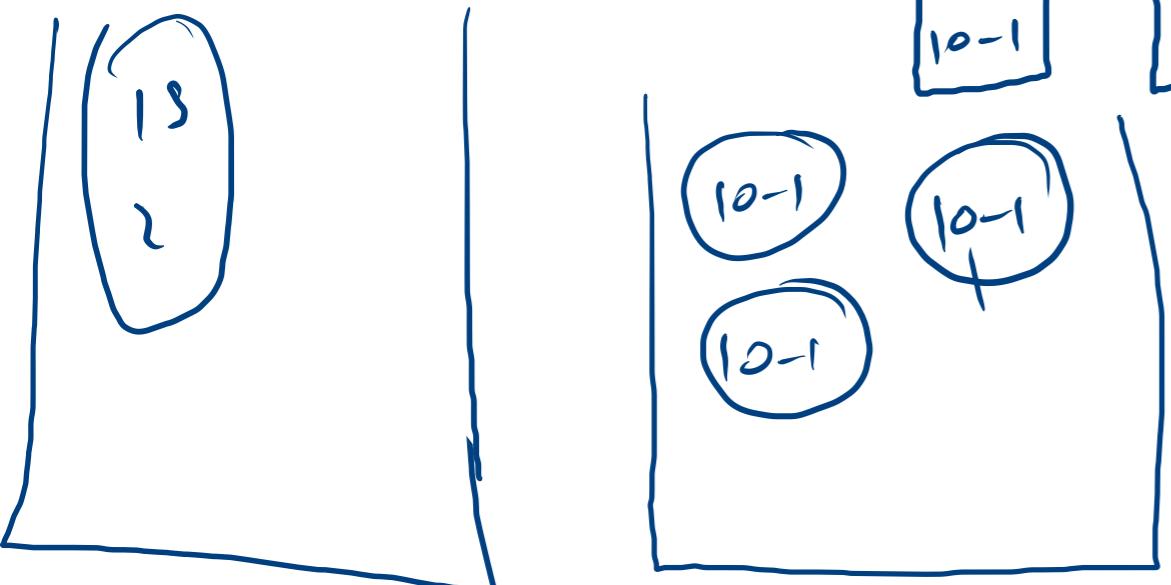
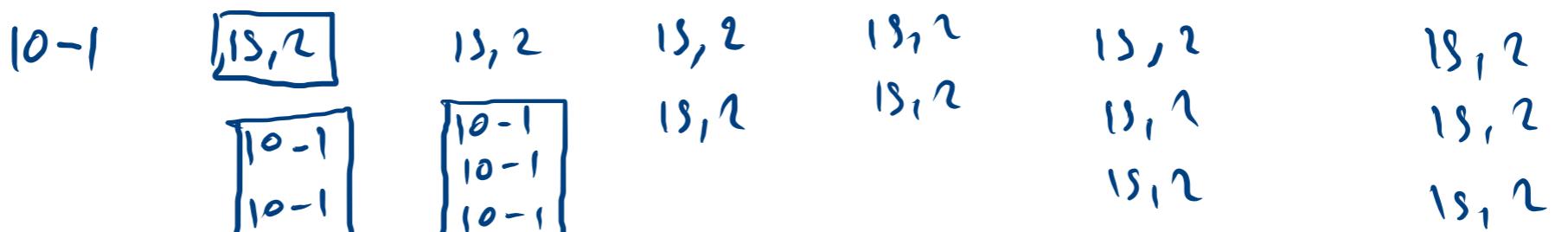
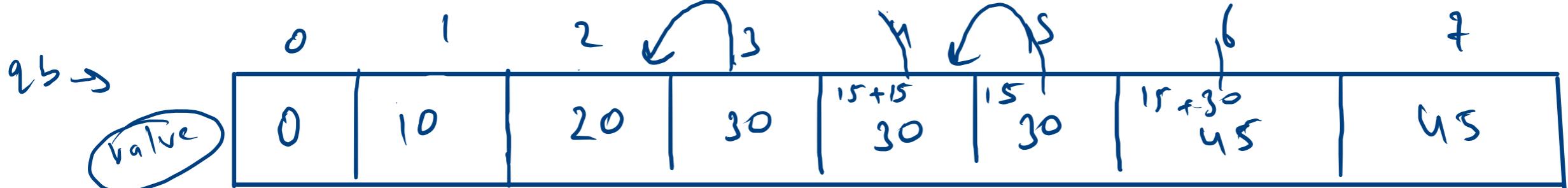
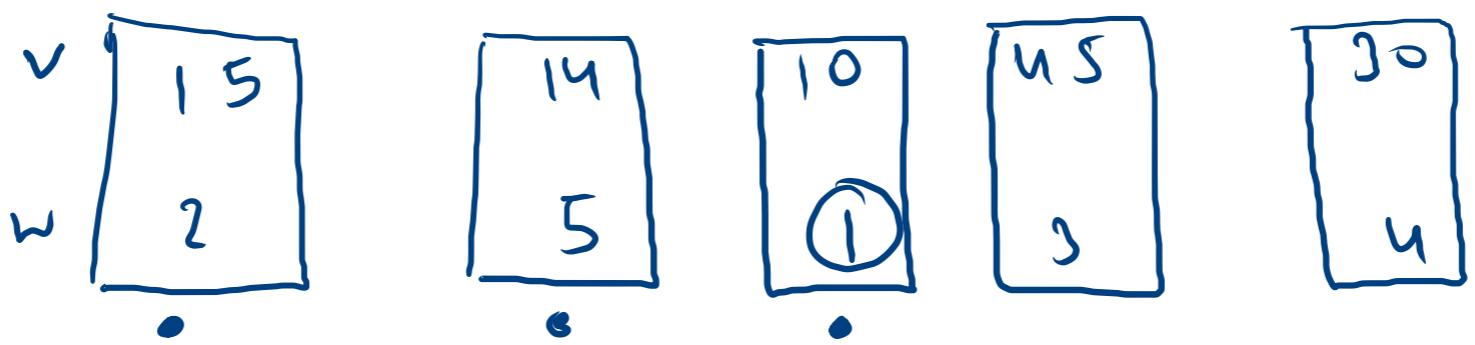
7 ←



depreciation



cap → 2



```

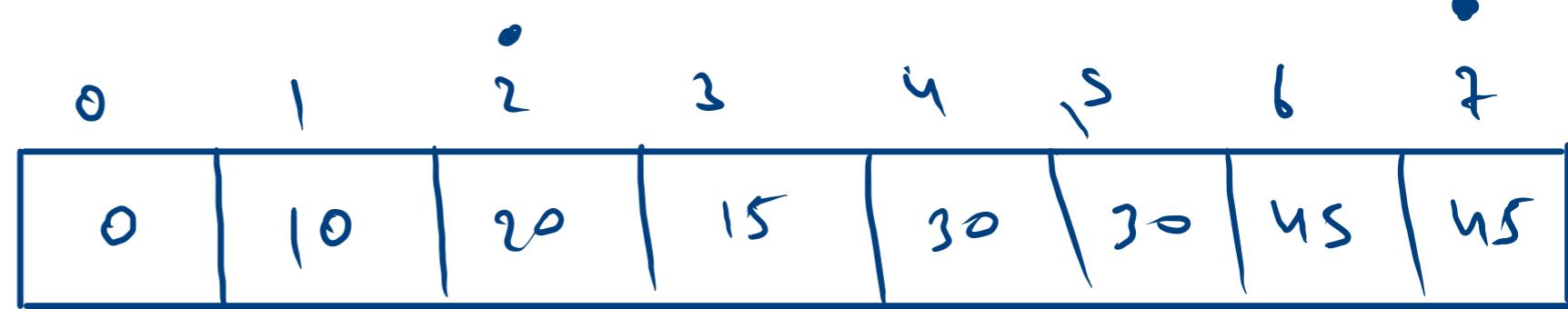
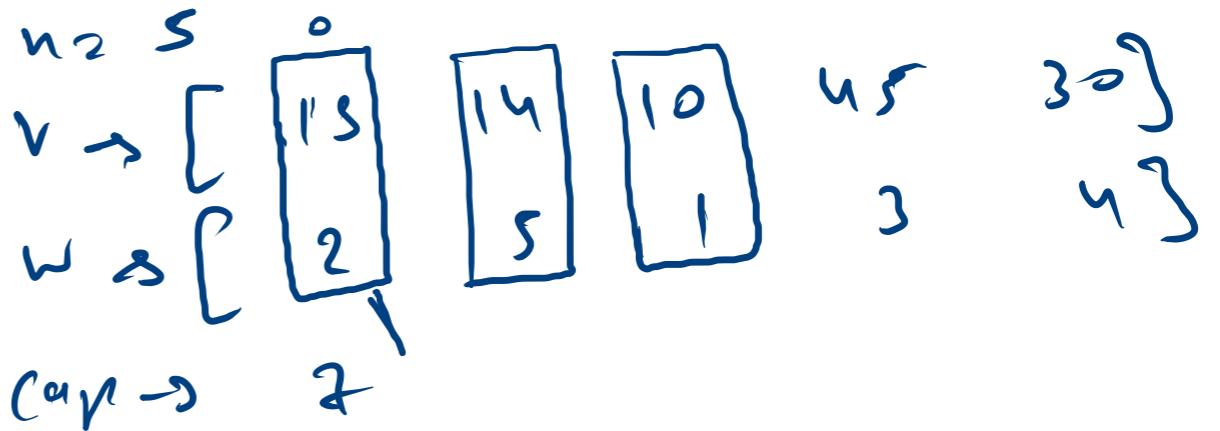
int qb[] = new int[cap+1];

for(int i=0;i<n;i++){
    int value = v[i];
    int weight = w[i];

    for(int c=weight;c<=cap;c++){
        int inc = value + qb[c-weight];
        if(inc > qb[c]){
            qb[c] = inc;
        }
    }
}

System.out.println(qb[cap]);

```



$int \rightarrow 0$
 $long \rightarrow 0$
 $double \rightarrow 0.0$
 $String \rightarrow null$

$h = 2, 3, 4, 5$

②

000

001

010 ✓

011 ✓

100

101 ✓

110 ✓

111 ✓



⑤

logic
↓
control
dynamic

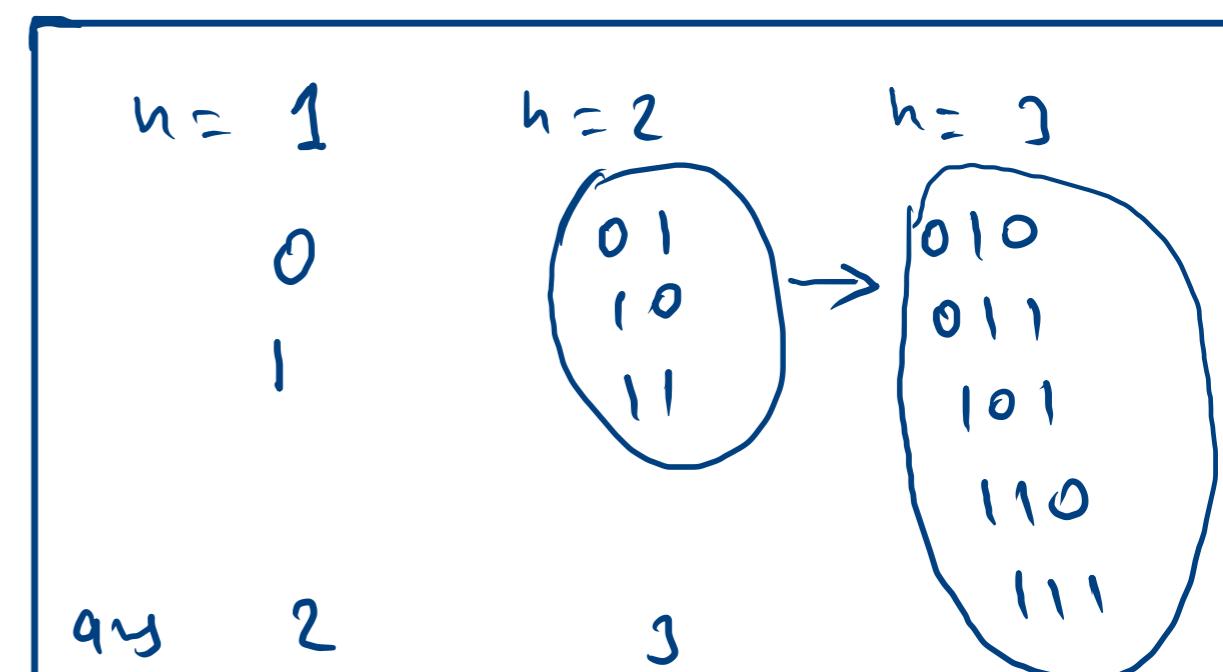
binary shr → '0' and '1'

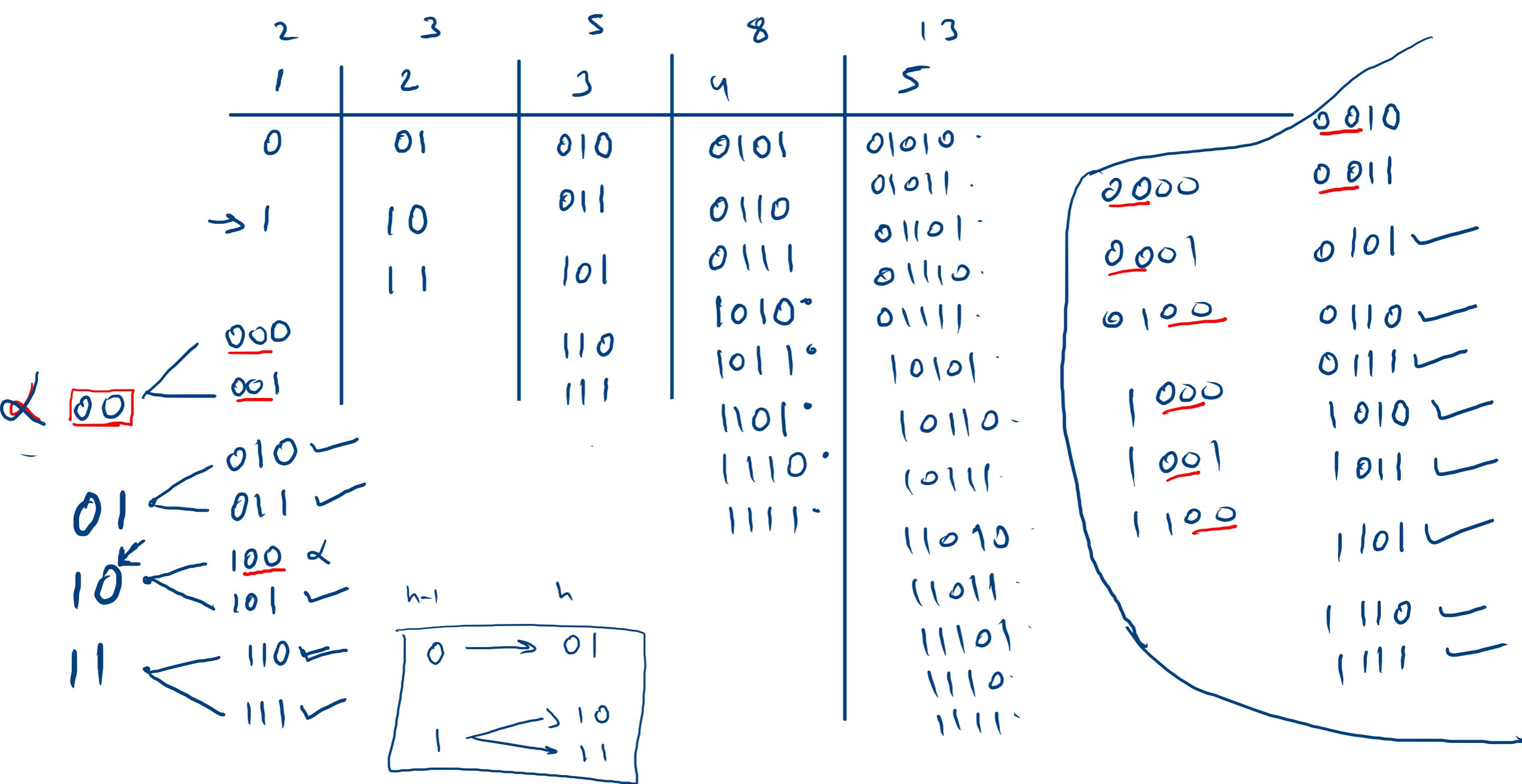
"00"

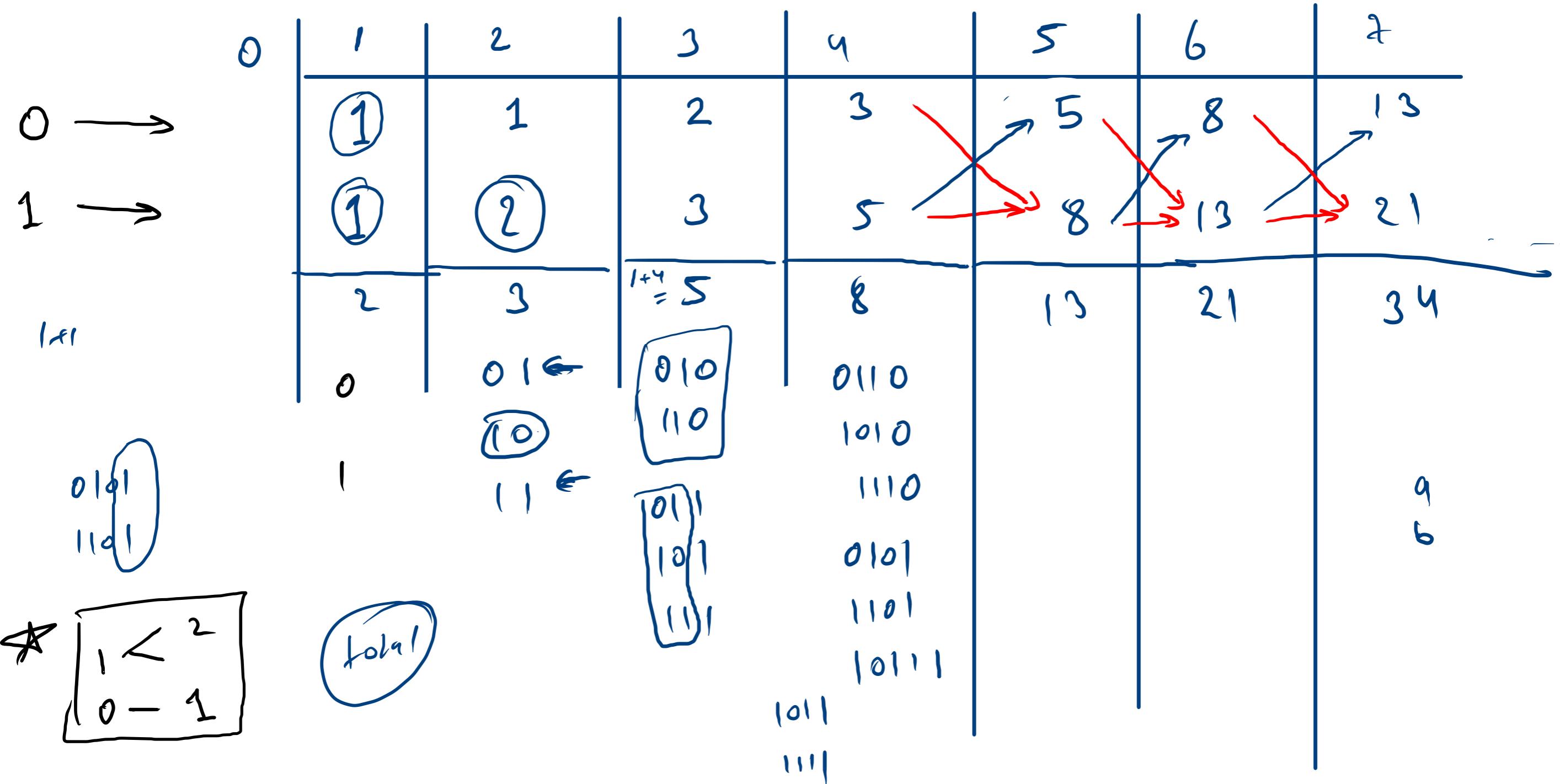
"11"

"010"

"100"







h2 3

(DD) 2

buildings

↓
B
S

space
J
S

B DD
D DS

t s

B	= S	'B
/ / / / / /		
0	e	o

B S B B S S S D S S S S S S S

D D D ✓
D S S ✓

t s

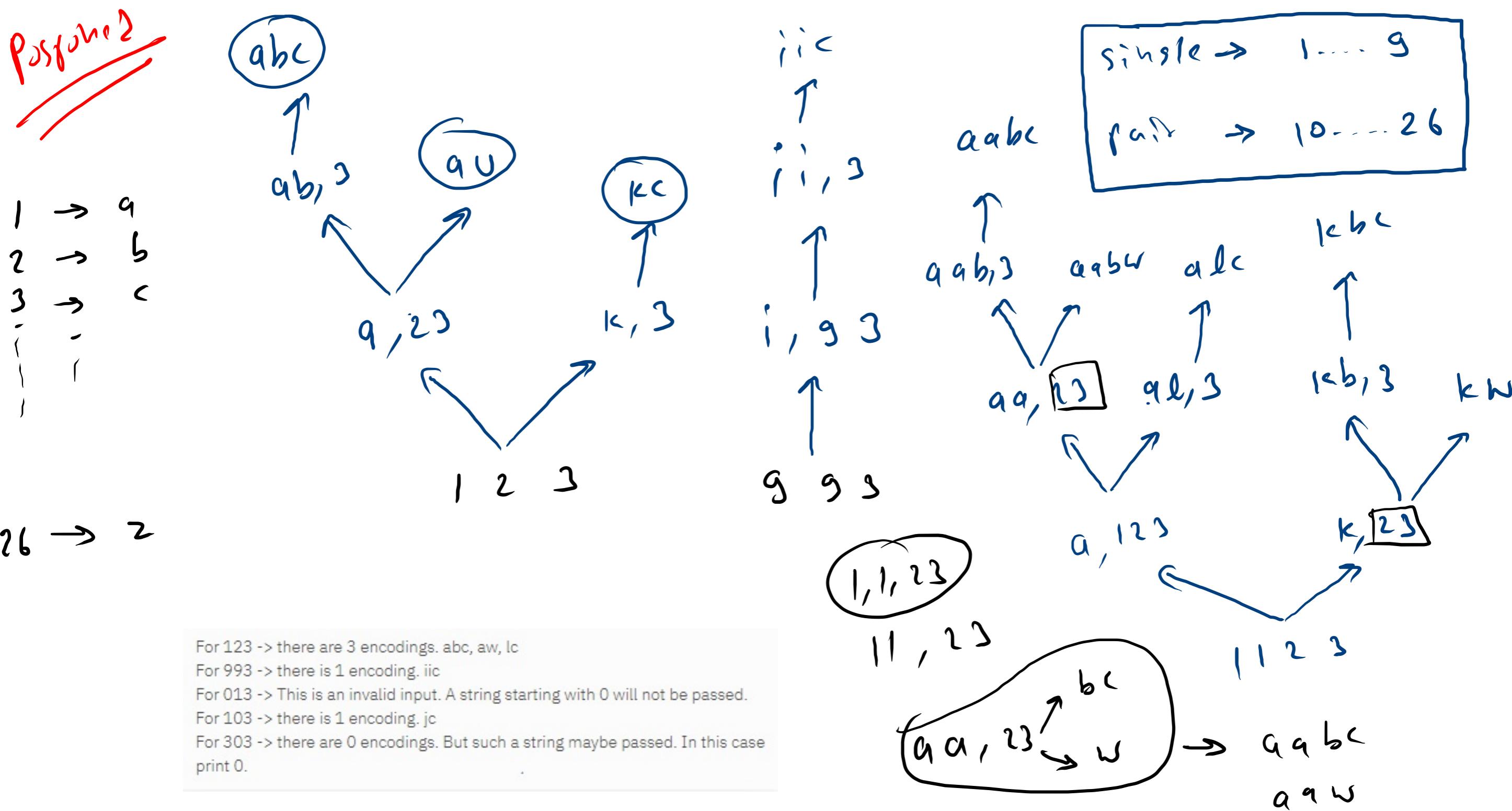
S DD
S D S ✓
S S D ✓
S S S ✓

t² 2S ↙

long ↙

double ↙ ← Math.pow()
int = int * int

B S B B S S S D S S S S S S S



~~Postponed~~

"11122"
π
0
1...5

