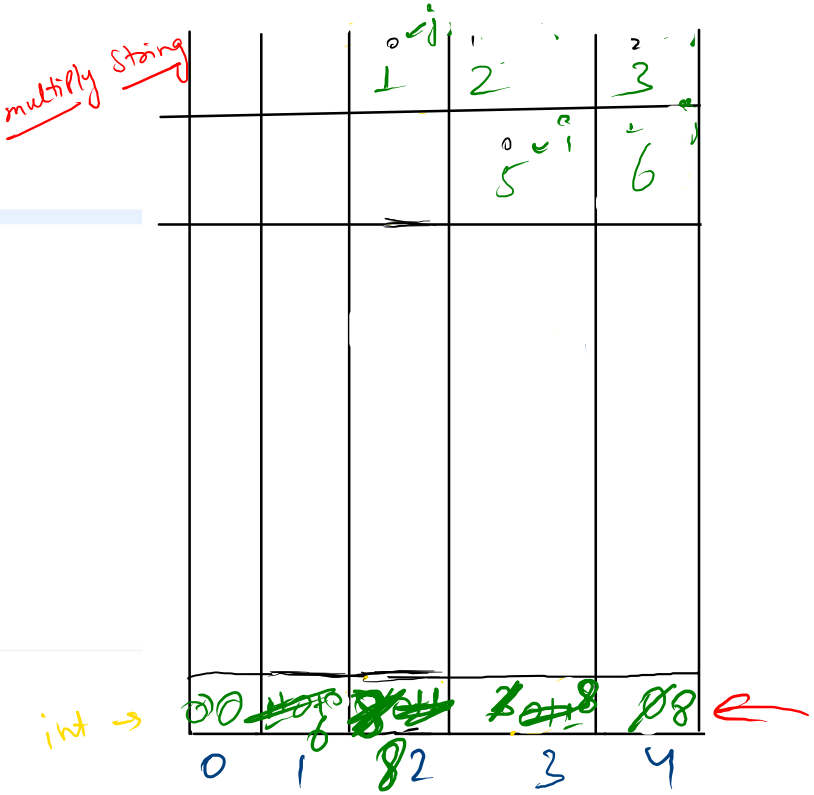
```java
public String multiply(String num1, String num2) {
    int[] ans = new int[num1.length()+num2.length()];

    for(int i= num1.length()-1;i>=0;i--){
        for(int j = num2.length()-1;j>=0;j--){

            int idx1 = i+ j;
            int idx2 = i+j+1;

            int val = (num1.charAt(i)-'0')*(num2.charAt(j)-'0');
            val = val + ans[idx2];

            int quo = val / 10;
            int rem = val % 10;

            ans[idx2] = rem;
            ans[idx1] += quo;

        }
    }

    StringBuilder sb = new StringBuilder();

    for(int i=0;i<ans.length;i++){
        if(sb.length()==0 && ans[i] == 0){
            continue;
        }

        sb.append(ans[i]);
    }

    if(sb.length()==0){
        return "0";
```



idx1 = i + j

idx2 = i + j + 1

idx1 = 0

idx2 = 1

val = 5 + 1 = 6

= 6

6 8 8 8 ⇒ 6888

max sum non-ovr

non-ovr

↳ all Possible L size m size subarrays

↓

$O(n)$

$L=3$
$m=2$

(I) m size prefinmax & Suffinmax

(II) L fin → m size

⇓

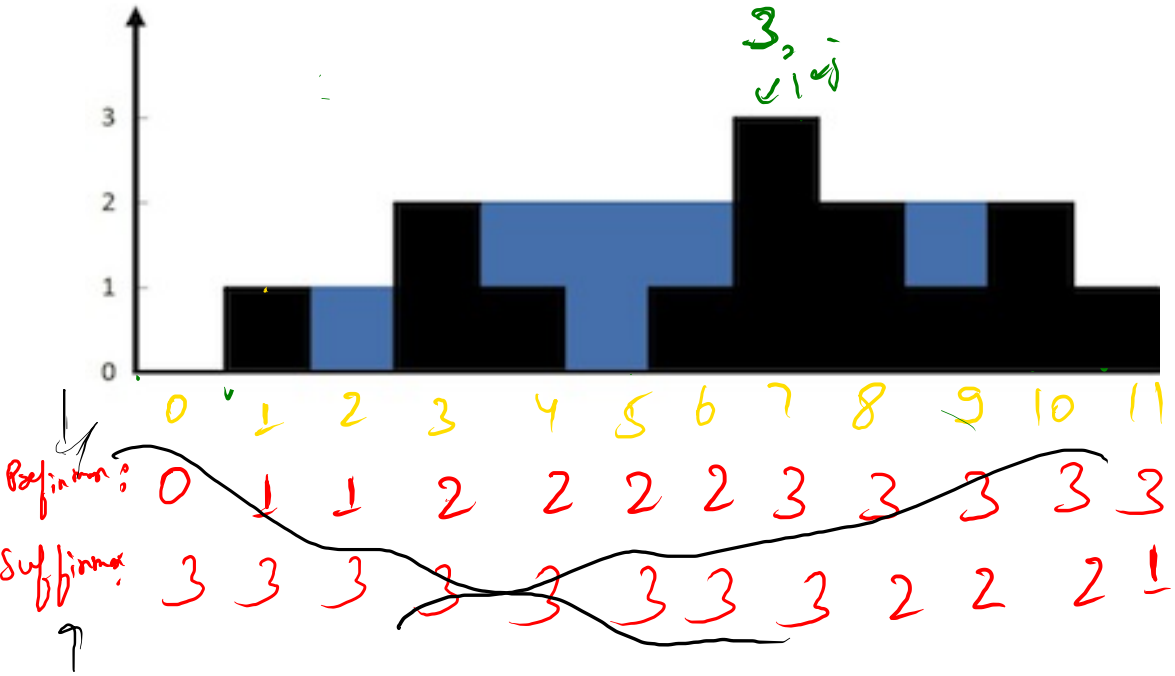
max

0	1	2	3	4	5
7	10	3	18	5	7
$-\infty$	17	13	21	23	12
17	13	21	23	12	$-\infty$

$-\infty$ 17 17 21 23 23

23 23 23 23 12 $-\infty$

0 max = $-\cancel{\infty}$ $\cancel{43}$ 47



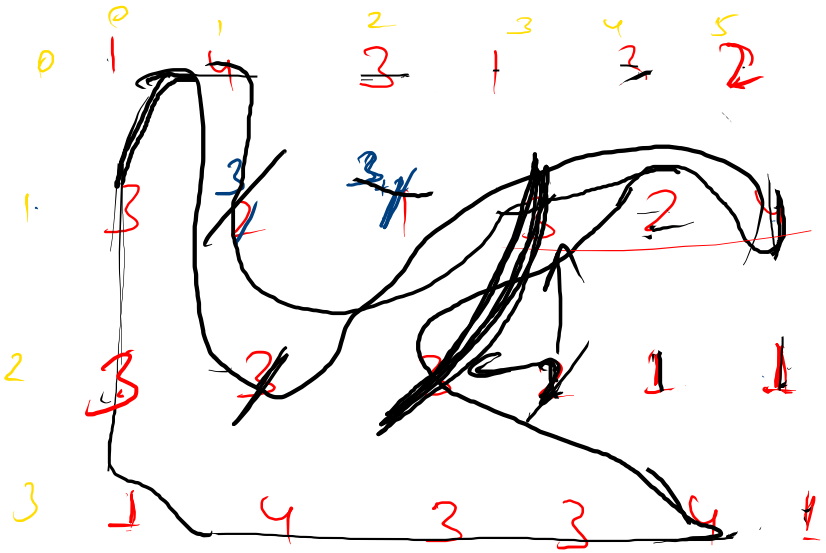
3, i, j

$\min(\text{Befinmax}, \text{Suffinmax}) - \text{height}[i]$

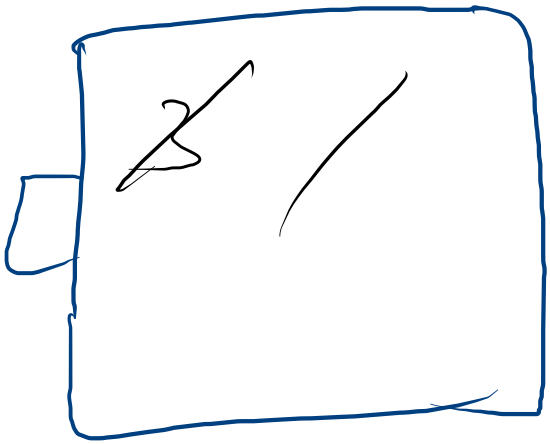
ans = 0 + 1 + 0 + 1 + 2 + 1 + 1 + 0

= 6

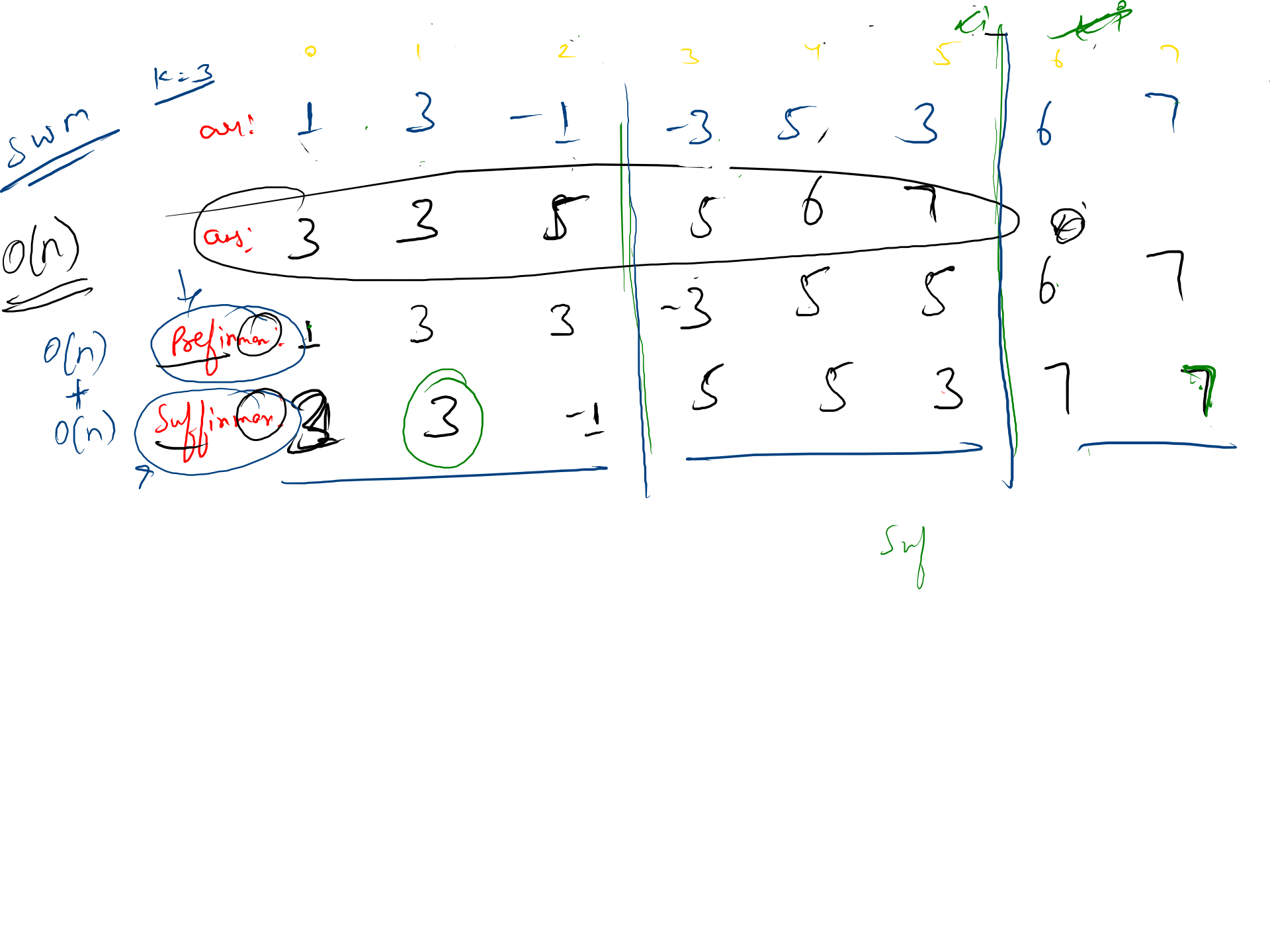
TRW-2



Pair (i, j, val)



only = 0 + 2 + 1


Swm
O(n)
k=3
0 1 2 3 4 5 6 7
arr: 1 3 -1 -3 5 3 6 7
ans: 3 3 5 5 6 7 6
Prefixmax
Suffixmax
O(n)
+
O(n)
1 3 3 -3 5 5 6 7
3 3 -1 5 5 3 7 7
Swf

Partition array

0	1	2	3			
5	8	9	3	12	14	16

Cmax = ~~5~~ 9

Omax = ~~5~~ ~~8~~ ~~9~~ ~~12~~ ~~14~~ 16

Pidx = ~~0~~ 3

left <= right

⇓

leftmax <= rightmin

Push dominoes

L . . L . R . . . L R . . L . . R

L L L . R

L . . . . L → L L L L L L L

R . . . . R → R R R R R R

L . . . . R → L . . . . R

R . . . . L → odd R R R . L L L

even R R R L L L