INFIX → POSTFIX

$3 + 4$ → $3\ 4\ +$

4×2 → $4\ 2\ \times$

$4 + \underbrace{3 \times 2}$ → $4\ 3\ 2\ \times\ +$

$\underbrace{4}\ \boxed{+}\ \underbrace{3\ 2\ \times}$

$\underbrace{4/3} \times 2 + 6 \times 3$ → $4\ 3\ /\ 2\ \times\ 6\ 3\ \times\ +$

$\underbrace{4\ 3\ /} \times \underbrace{2} + 6 \times 3$

$\underbrace{4\ 3\ /\ 2\ \times} + \underbrace{6\ 3\ \times}$

$4\ 3\ /\ 2\ \times\ 6\ 3\ \times\ +$

Q Given postfix expression. Evaluate it!

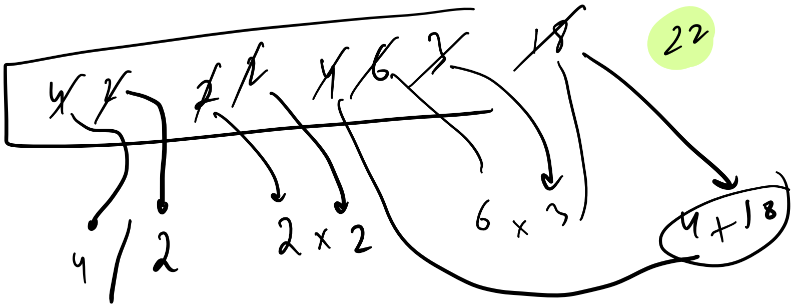
$\boxed{\boxed{4\ 3\ +}\ \boxed{\boxed{6\ 3\ \times}\ \boxed{4\ 2\ +}\ /}\ \times}$ ⇒ $(4 + 3) \times ((6 \times 3)/(4+2))$

$\underbrace{4\ 3\ +}\ \underbrace{6\ 3\ \times}\ \underbrace{4\ 2\ +}\ /\ \times$

~~7~~ ~~18~~ ~~6~~ ~~3~~ 21

DS : Stack

4 2 / 2 x 6 3 x + 3 2



I Operand

push it in a stack

II operator

op2 → pop one item from stack

op1 →

push in stack

op1 operator op2

ANS : Only item in the stack .

$(3+4) \times 10 \rightarrow$ 3 4 + 10 ×

evaluate

Q Given an array. Find the nearest smaller element on left of every element.

A:	4	5	2	10	8	2
NSL:	-1	4	-1	2	2	-1

A :	4	6	10	11	7	8	3	5
NSL :	-1	4	6	10	6	7	-1	3

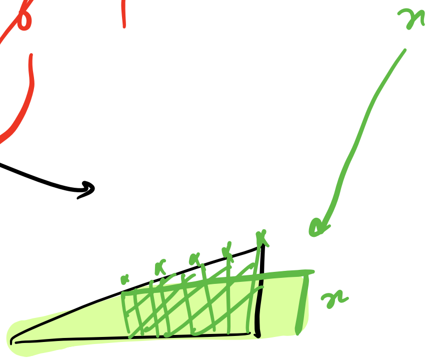
1) BF

```
f (i: 0 → N-1) {
    f (j: i-1 → 0)
        if (A[j] < A[i])
```

$\boxed{TC = O(N^2)}$

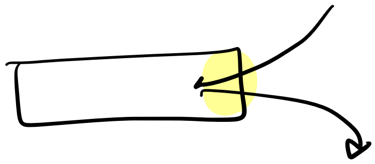
$\boxed{SC = O(1)}$

A :	5	2	8	10	12	6	1
NSL :	-1	-1	2	8	10	2	-1
CL :	~~5~~	~~2~~	~~8~~	~~10~~	~~12~~	~~6~~	1



Requirements of DS [CL] ?

1) Insert at END
2) Remove elements from end
3) Read element at that end
4) Maintain insertion order



STACK

```
CODE :    // A[N]
 stack <int> CL;
 ANS [N];
 f(i=0; i<N; i++) {                                N
     while (! CL.is empty() && CL.top() >= A[i]) {
         CL.pop();
     }
     if (!CL.is empty())
         ANS[i] = CL.top();
     else ANS[i] = -1;
     CL.push(A[i]);
 }
 ret ANS[];
```

DEL : ≤ N
CL
INS : N

$\boxed{TC = O(N)}$

$\boxed{SC = O(N)}$

WC for space

1	2	3	4	5
-1	1	2	3	4

CL : 1 2 3 4 5

Q Given an Array. for every element find the indent of the Nearest Smaller element on left → NSL

	0	1	2	3	4	5
A :	2	7	6	3	1	5
NSLi →	-1	0	0	0	-1	4

CL : [0 ~~1~~ 2 . — —
[idx]

CODE : A[N]

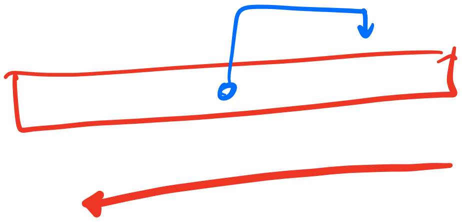
```
stack <int> CL;
ANS[N];
f(i=0; i<N; i++) {
    while(!CL.isEmpty() && A[CL.top()] >= A[i]) {
        CL.pop();
    }
    if(!CL.isEmpty())
        ANS[i] = CL.top();
    else ANS[i] = -1;
    CL.push( i );
}
ret ANS[];
```

TC = O(N)

SC = O(N)

Q Given A[].
find NS on Right ?

NSR



CODE : A[N]

```
stack <int> CL;
ANS[N];
f(i : N-1 ; i >= 0 ; i--) {
    while(!CL.isEmpty() && A[CL.top()] >= A[i]) {
        CL.pop();
    }
    if(!CL.isEmpty())
        ANS[i] = CL.top();
    else ANS[i] = -1;
    CL.push( i );
}
ret ANS[];
```

TC → O(N)
SC ↗

Q   Given   A[].   find   NGL . ?

greater

CODE :   A[N]

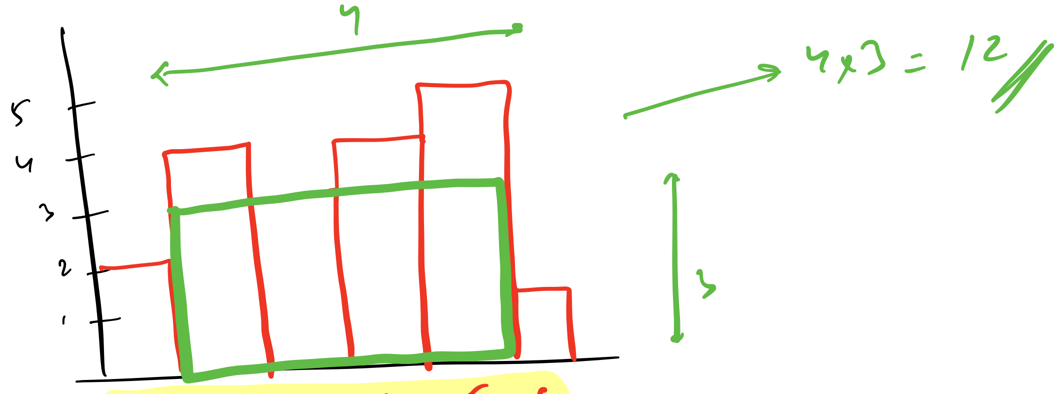
```
stack <int> CL;
ANS[N];
f(i=0; i<N; i++) {
    while(!CL.isEmpty() && A[CL.top()] <= A[i]) {
        CL.pop();
    }
    if(!CL.isEmpty())
        ANS[i] = CL.top();
    else ANS[i] = -1;
    CL.push( i );
}
ret ANS[];
```

Q   NGR ?

SAME AS ABOVE

f.(N-1 ⟶ 0)

Q Given continuous blocks of histogram.
Find the maximum rectangular area.



A : 2 4 3 4 5 1

1) BF

find the S.A to choose as width → N^2

→ [L,R]

h → min(A[L] -- A[R]) → N

x

w → R-L+1

MAX ←

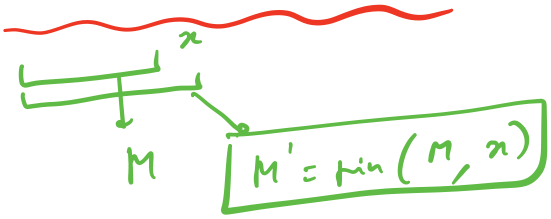
↓

ANS

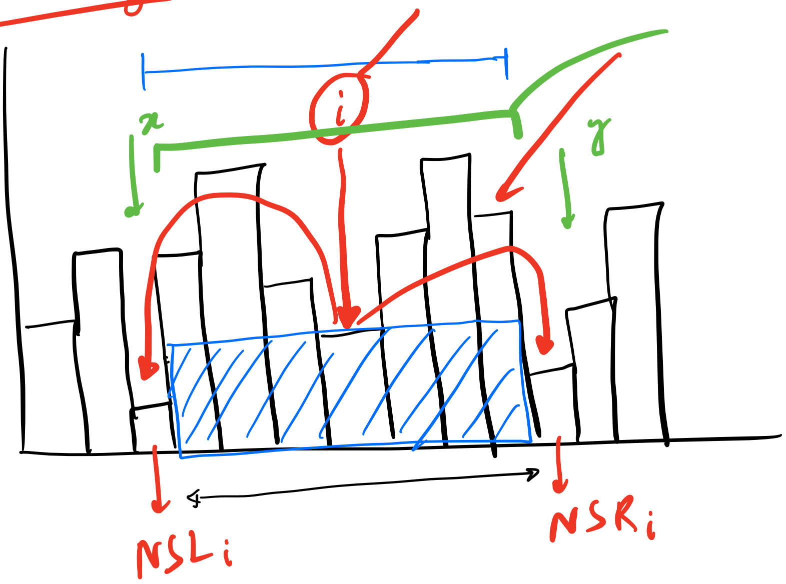
$\boxed{TC: O(N^3)}$

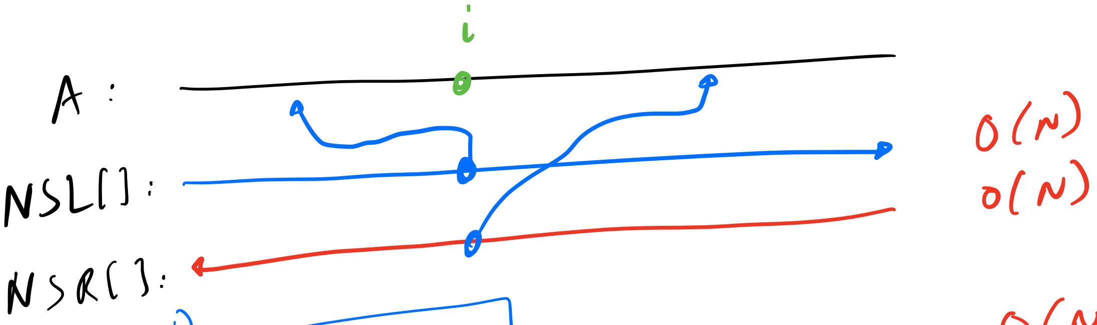
2) Carry forward

$\boxed{TC: O(N^2)}$



3) fin the height





O(N)

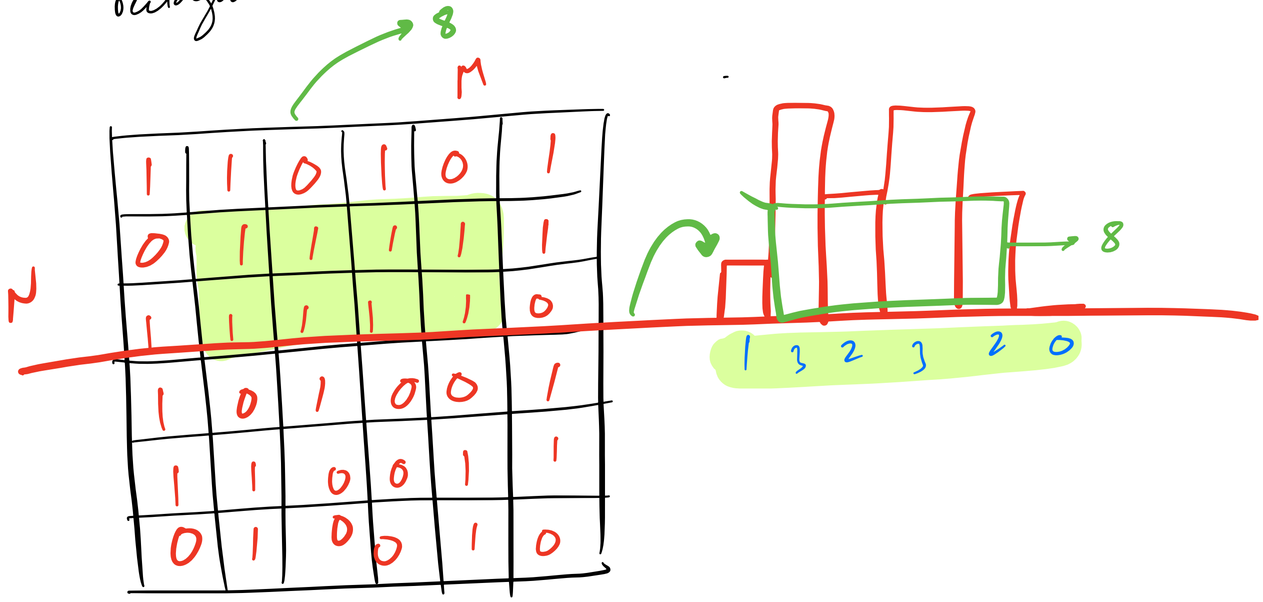
```
f(i:0 → N-1) {
    h → A[i]
    w → NSR[i] - NSL[i] - 1

    Area = h x w
    ANS = MAX(ANS, Area);
}
```

$\boxed{TC = O(N)}$

$\boxed{SC = O(N)}$

Q Given a binary Matrix. Find the MAX rectangular area with all 1's.



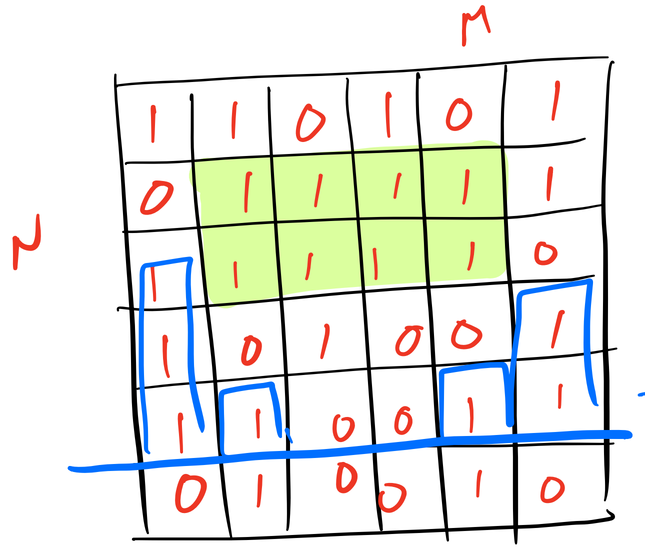
1	1	0	1	0	1
0	1	1	1	1	1
1	1	1	1	1	0
1	0	1	0	0	1
1	1	0	0	1	1
0	1	0	0	1	0

1) BF

$\#SM = N^2 M^2$

$O(1)$ PS

$$\boxed{TC : O(N^2 M^2)}$$



1	1	0	1	0	1
0	1	1	1	1	1
1	1	1	1	1	0
1	0	1	0	0	1
1	1	0	0	1	1
0	1	0	0	1	0

Colwise ps :

1	1	0	1	0	1
0	2	1	2	1	2
1	3	2	7	2	0
2	0	3	0	0	1
3	1	0	0	1	2
0	2	0	0	2	0

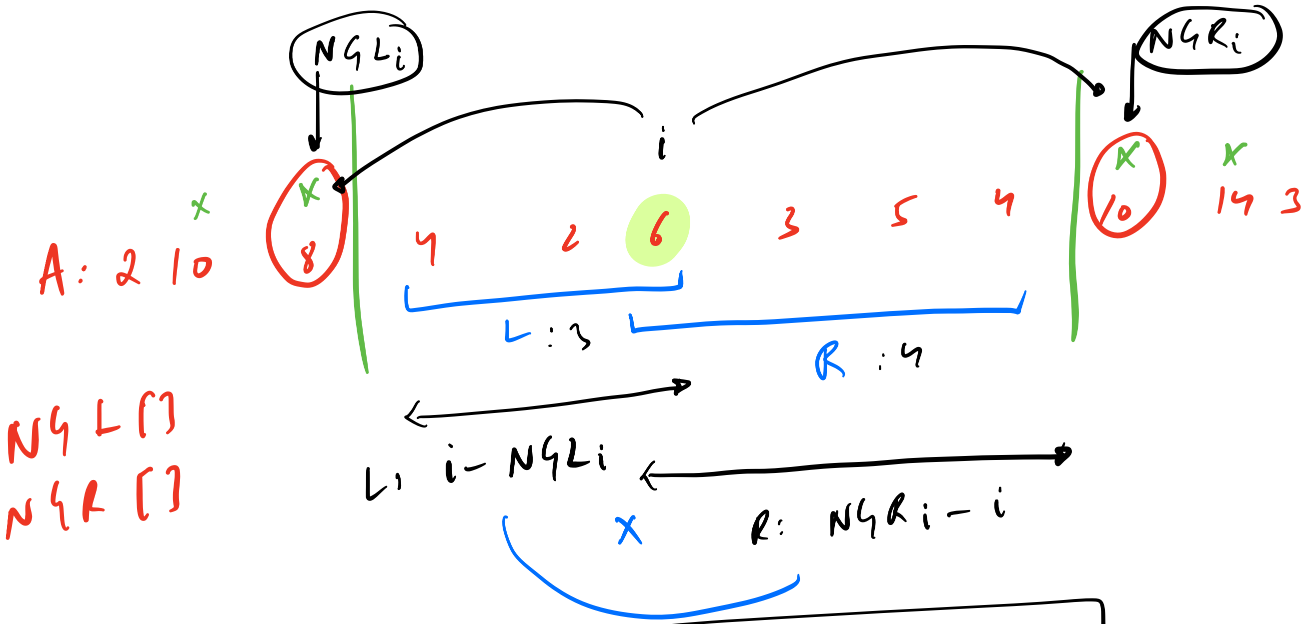
Q Given an array. find the sum of MAX of all SAs.

A : [1, 4, 3]

SA	MAX
(1)	1
(1, 4)	+ 4
(1, 4, 3)	+ 4
(4)	+ 4
(4, 3)	+ 4
(3)	+ 3
	20

Contribution of every element

element		#SAs		
1	×	1	=	1
4	×	4	=	+ 16
3	×	1	=	+ 3
				20



$$\boxed{\#SA \text{ with } A(i) \text{ as MAX} = (i - NGL_i) \times (NGR_i - i)}$$

$$\boxed{\text{Contribution by } A[i] = (i - NGL_i) \times (NGR_i - i) \times A[i]}$$

$$ANS = \sum_{i=0}^{N-1}$$

$$\boxed{TC = O(N)}$$

$$\boxed{SC = O(N)}$$

	0	1	2	3	4	5	6	7	8
A :	1	2	3	4	5	4	3	2	1
NGL :	-1	-1	-1	-1	-1	4	5	6	7
NGR :	1	2	3	4	9	9	9	9	9

R : 1

4 : L