

Infix, Postfix and Prefix

Infix

We generally write infix expressions, used in mathematical expression, used in program that we will.

→ Infix Operation

Infix	Postfix	Prefix
$x+y$	$xy+$	$+xy$

→ Advantages of Prefix & Postfix

- Don't require parenthesis, precedence rules & associativity rules.
- Can be evaluated by writing a program that traverses the given expression exactly once. (In Infix more than one pass required)

Operations	Associativity	precedence
\wedge	R to L	
$*, /$	L to R	
$+, -$	L to R	

Example:	Infix	Prefix	Postfix
(i)	$x + y * z$	$+x +yz$	$xyz + *$
(ii)	$(x+y) * z$	$*+xyz$	$xyz + *$

When converting Infix to Postfix or Prefix we need associativity Rule & precedence rule

Step

Prefix

Steps for Postfix Conversion

(2) Inside the parenthesis expression

$$\begin{array}{l} \text{(i)} \quad x+y*z \quad \text{Postfix} \\ (x + (y * z)) \\ (x + (yz *)) \\ xyz * + \end{array}$$

$$\begin{array}{l} \text{Postfix} \\ x+y+z \\ (x + (y + z)) \\ (x + (+yz)) \\ +xyz \end{array}$$

$$\begin{array}{l} \text{(ii)} \quad (x+y)+z \quad \text{Postfix} \\ ((x+y)+z) \\ ((xy)+*z) \\ xy+*z \end{array}$$

$$\begin{array}{l} \text{Postfix} \\ (x+y)+z \\ ((x+y)*z) \\ ((+xy)*z) \\ *xyz \end{array}$$

→ Infix to Postfix (conversion)

(a) Naive (Same as above)

$$\begin{array}{l} a+b*(c-d) \\ (a+(b*(c-d))) \\ (a+(b*(cd-))) \\ (a+(bcd-*)) \\ (a+ abcd-*+) \end{array}$$

$$\begin{array}{l} a+b*c/d+e \\ ((a+((b*c)/d))+e) \\ ((a+((bc*)/d))+e) \\ ((a+(bc*d/))+e) \\ ((abc+d/+)+e) \\ abc+d/+e+ \end{array}$$

b) Algorithm $O(n)$ $S \rightarrow O(n)$

① Create an empty stack, st

② Do following for every character x from left to right.

③ If $x \in b$:

a) Operand : output it

'C' b) Left Parenthesis : Push to st

c) Right _____ : pop from st until left parenthesis is found. Output the popped operators.

d) Operator:

If st is empty, push x to st
else compare with st top.

(i) Higher precedence (than st top), push to st.

(ii) lower _____, pop st top and output until a higher precedence operator is found & then push s to st.
on 'C' is found

iii) Equal precedence, use associativity output everything from st. f

Implementation

```
string f(string s)
{
    stack<char> st;
    string sl = " ";
    for (int i = 0; i < s.length(); i++)
    {

```

$y(s[i]) \geq 97 \text{ & } s[i] < 122$

$st[i] = sl + s[i];$

else $y(s[i]) == '('$

$st.push(s[i]);$

else $y(s[i]) == ')'$

```
{ while (!st.empty() && st.top() != '(')
```

```
{     sl += st.top();
          st.pop();
      }
```

```
} st.pop();
```

else

```

    {
        while (!st.empty() && st.top() != 'C' &&
               brc(s[i]) <= brc(st.top()))
        {
            st += st.top();
            st.pop();
        }
        st.push(s[i]);
    }
}

```

while (!st.empty())

```

    {
        st += st.top();
        st.pop();
    }
}

```

return st;

removing remaining
characters from stack

int brc(char c)

{

if (c == '^')

when 3;

else if (c == '*' || c == '/')

when 2;

else

when 1;

precedence

Evaluation of Postfix

Ex-43 v = ["4", "13", "5", "/", "+"]

$$(4 + (13 / 5)) = 6$$

```
int } (vector<string>& v)
```

```
{ stack<int> st;
```

```
int for(int i=0; i < v.size(); i++)
```

```
{ if(v[i] == "+" || v[i] == "-" || v[i] == "*" || v[i] == "/")
```

```
int a = st.top();
```

```
st.pop();
```

```
int b = st.top();
```

```
st.pop();
```

```
if(v[i] == "+") st.push(a+b);
```

```
else if(v[i] == "-") st.push(b-a);
```

```
else if(v[i] == "*") st.push(a*b);
```

```
else st.push(b/a);
```

```
else
```

```
st.push(stoi(v[i]));
```

```
} when st.top();
```

String to integer

Remember, v[i] (" ") double quote it ~~is not - it is~~ no character
 यह नहीं है अक्षर (like '2' X) "2"

→ Infix to Prefix

a) Naive

first fully parenthesis it

start solving from innermost to outermost

$$(x+y) * (z+w)$$

$$((x+y) * (z+w))$$

$$(+xy) + (+zw) \rightarrow *+xy+zw$$

$$x+y+z/w+u$$

$$((x + ((y + z)/w)) + u)$$

b) Algorithm

• make an empty stack, st
• make — str, str

• Do following on every character ch from right to left, traversal

• If ch is

a) operand : push it to str

b) Right parenthesis ()): push to st

c) Left parenthesis ('(') : pop from st until right parenthesis is found. Append the popped characters to prefix str.

d) operator :

i) If st is empty, push ch to st
else compare with st top.

(i) Higher precedence (than st top) push ch to st

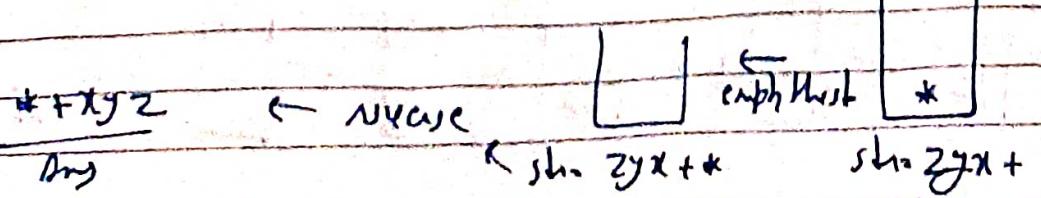
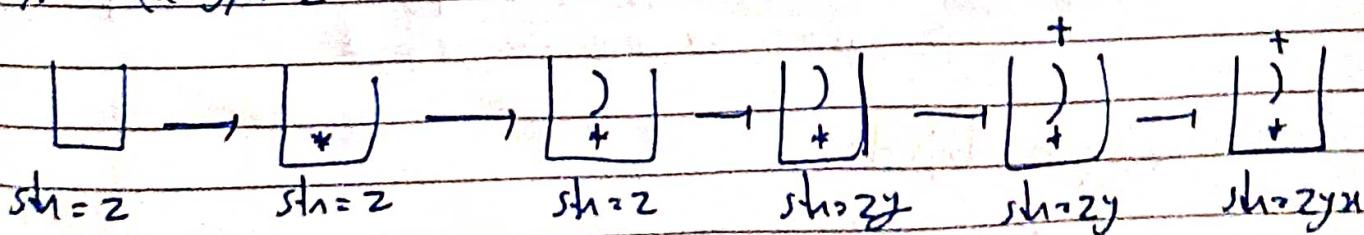
(ii) lower — : pop st top and append the popped item to str until a higher precedence operator is found (or st becomes empty) push ch to st.

(iii) Equal precedence : Use Associativity.

→ Now pop every item from st & append to str

→ Now reverse str to get answer

$$q_1: (x+y)+z$$

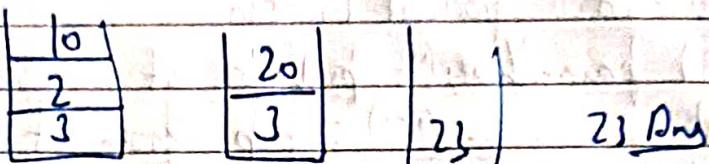


→ Evaluation of Prefix

$\text{++ } 10 \ 2 \ 3$

Just traverse from right to left \leftarrow , where push in stack every number whenever operator (+) encountered pop() from top of stack and do operation & push the result into stack.

e.g. $\text{++ } 10 \ 2 \ 3$



★ ★ Largest Rectangle in Histogram

$O(n) \dots O(n)$

```
int f(vector<int> &v)
```

```
{ int n = v.size();
```

```
stack<int> st;
```

```
int ans = 0, curr = 0;
```

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

```
{
```

2nd while loop start
ans

```
while (!st.empty() && (i == n || v[st.top()] >= v[i]))
```

```
{
```

```
int ind = st.top();
```

```
st.pop();
```

```
curr = v[ind] + (st.empty() ? i : (i - st.top() - 1));
```

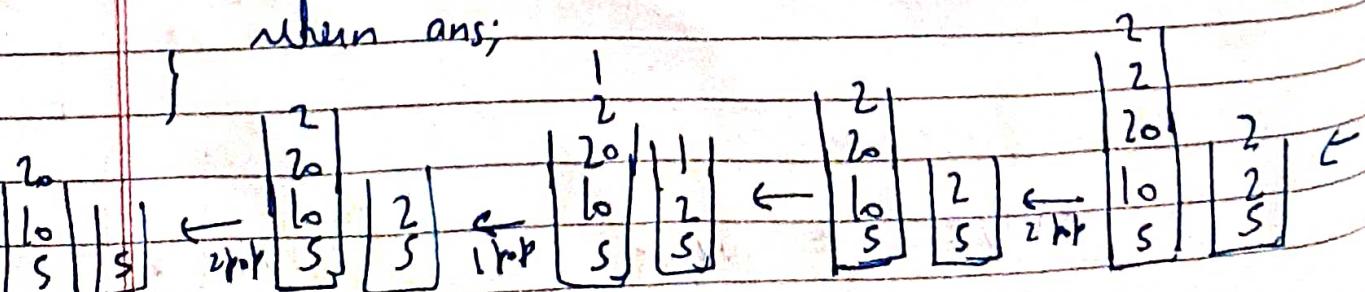
```
ans = max(ans, curr);
```

```
}
```

```
st.push(i);
```

```
}
```

```
return ans;
```



एक stack और vector का implement करते हैं तो
यह तो O(n) sc की complexity (S1 वला)

class AuxStack
Data _____
Page _____

SC

1

→ Design a stack that supports getmin() $O(1)$ $O(n) \rightarrow S2$
vala

a) Using Two Stack

stack<int> s1, s2;

void push(int val)

{ s1.push(val); }

i) (s2.empty() || s2.top() \geq s1.top())

s2.push(val);

}

void pop()

{ if (s1.top() == s2.top())

s2.pop();

s1.pop();

}

int top()

{ return s1.top(); }

int getMin()

{ return s2.top(); }

- We maintain an auxiliary stack (S2 here) we maintain the min element in this stack.
- We only push to S2 when $s2.top() >= s1.top()$

e.g. push(5) push(10) push(2) push(2)
push(6) push(4) pop() pop()
push(2) pop() push(1)
pop() pop()

equally b/w duplicates no
present

getmin() always give min
from s2.top()!

5		2	2	4	2	2	2	2
		10	5	6	5	4	6	5
			2	2	2	2	2	2
				10	5	4	6	5
					5	4	6	5

valid for -ve no. also



b) stack<long long int> s;
 long long int min;

$T(-O(1))$

for every operator

$SC \sim O(1)$

void push(int val)

{ long long int val = (long long int)x;

y(s.empty())

{ s.push(val);

min = val;



else if (val <= min)

{ s.push(2 + val - min);

min = val;



else

s.push(val);

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

<div data-bbox="162 3584 180 3590" style

Hard

Q) Maximum or minimum for every window size

[10, 20, 30, 50, 10, 20, 30]

o/p 20 30 20 10 10 10 10

वह observation o/p sorted 3D दृष्टि

Area of histogram

it related to

vector<int> f(int a[], int n)

{ vector<int> v(n, 0);

stack<int> st;

} for(int i=0; int i=0; i<=n; i++)

while(!st.empty() && (i==n || a[st.top()]>=a[i]))

{

int t = st.top();

st.pop();

int ind;

ind = st.empty() ? i : i-st.top()-1;

v[ind-i] = max(v[ind-i], a[t]);

}

st.push(i);

}

} for(int i=n-2; i>0; i--)

{ if(v[i]<v[i+1])

v[i] = v[i+1];

soho

20 50 30 20 0 0 0 10

20 + length of min len

at 6 length of 2 = 10

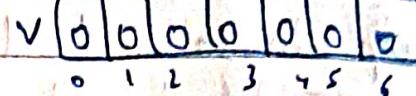
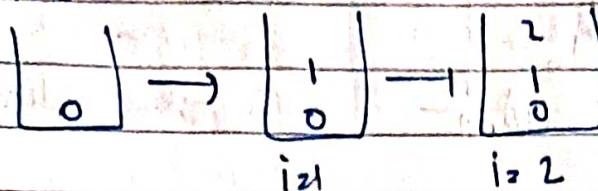
when v:

जब जो pop हो एवं element वह answer का हो तो window length

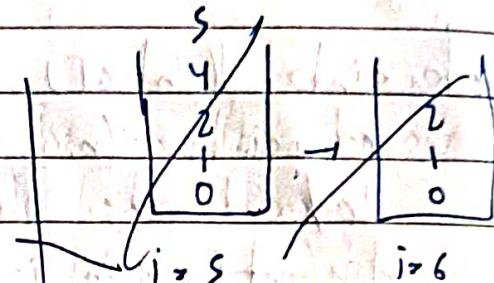
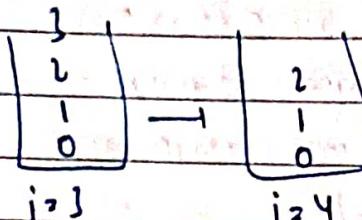
0 1 2 3 4 5 6

g1 [10, 20, 30, 50, 10, 20, 30]

i=0



Initially



start bpb

while loop \rightarrow t=3
ind = 4 - 2 - 1 = 1

$v[0] = \max(v[0], a[3])$

$v[0] = 50$

start bpb(i)

while loop continues
t=2

$ind = 4 - 1 - 1$

$= 2$

$v[1] = \max(v[1], a[2])$
 $= 30$

while loop continues

$t=1$

start bpb(i)

$ind = 4 - 0 - 1 = 3$

$v[2] = \max(v[2], a[1])$
 $= 20$

while loop cont.

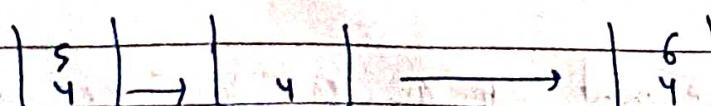
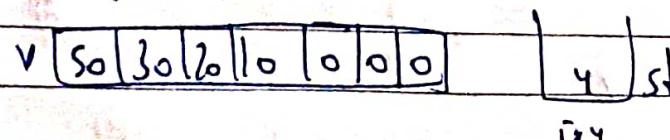
$t=0$

start bpb(i)

$ind = 4 - 0 - 0 = 4$

$v[3] = \max(v[3], a[0])$
 $= 10$

after while loop ends



$t=5$

start bpb(i)

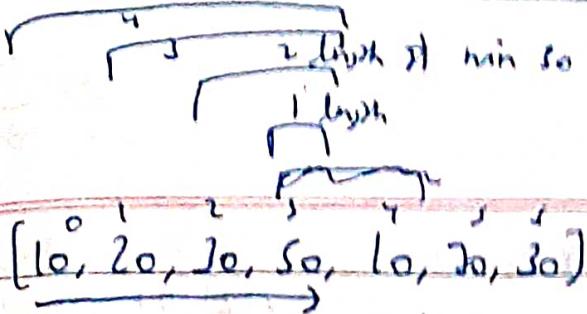
$ind = 6 - 4 - 1 = 1$

$v[0] = \max(v[0], a[5])$

$= 20$

similar others

(i=5) start bpb(i)



\Rightarrow अब इसे, जिस min element वाली सूची का length
को stack में वही element जो min(a[s.top()]) है उस
length का min बढ़ावा

③ Remove K digit The Celebrity Problem $O(n) \ O(1)$

$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	if $av[i][j] == 1$ then i knows j
	$\Rightarrow v[i][i] == 0$ for all i

Celebrity is a person whom everyone knows but he doesn't know anyone.

```

int f(vector<vector<int>> v, int n)
{
    int c=0; // let c be celeb
    for(int i=1; i<n; i++)
        if(v[c][i]==1)
            c=i;
    for(int i=0; i<n; i++)
        if((i!=c) && (v[i][c]==0 || v[c][i]==1))
            return -1;
}
  
```

} Jind the candidate for celeb

return c;

$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 2 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0 \end{bmatrix}$	$c=0$
---	-------

मगर $v[0][1]==1$
0-1 नहीं पाना है
तो 1 celebrity ही पाना है और 0 नहीं पानी
0 पाना है तो position = c=1

$$v[1][2]==1$$

$\Rightarrow 1 \rightarrow 2$ नहीं पाना है $\therefore 1$ can't be celeb $\because 2$ can be $\therefore c=2$

$v[2][3]==0$ $2 \rightarrow 3$ X नहीं पाना, i=3 stop Now let's find answer $c=2$

for i=2 to b1 . all bi's

v[0][2]

v[1][2]

v[2][2]

v[2][0]

v[2][1]

v[2][3]

} all should
be zero.

(*)

Remove k digits

Given string & index k, return the smallest possible index after removing k digits from string.

- Logic: smallest no banana is it obvious but if left if smallest no. digit not st.top() > num[i] not remove

"1432219"
op "1219"

k=3

"10200"
k=1

"200" (leading zero not remove)

string f(string num, int k)

{ stack<char> st;

int c=0;

if (num.size() <= k) return "0";

for (int i=0; i < num.length(); i++)

{ while (!st.empty() && st.top() > num[i] && c < k)

st.pop();

c++;

st.push(num[i]);

string s;

while (c != k)

{ st.pop(); c++; }

discusses

Date _____

Page _____

while(!st.empty())

{
 s = st.top();
 st.pop();
}

for(int i=s.length()-1; i>=0; i--)

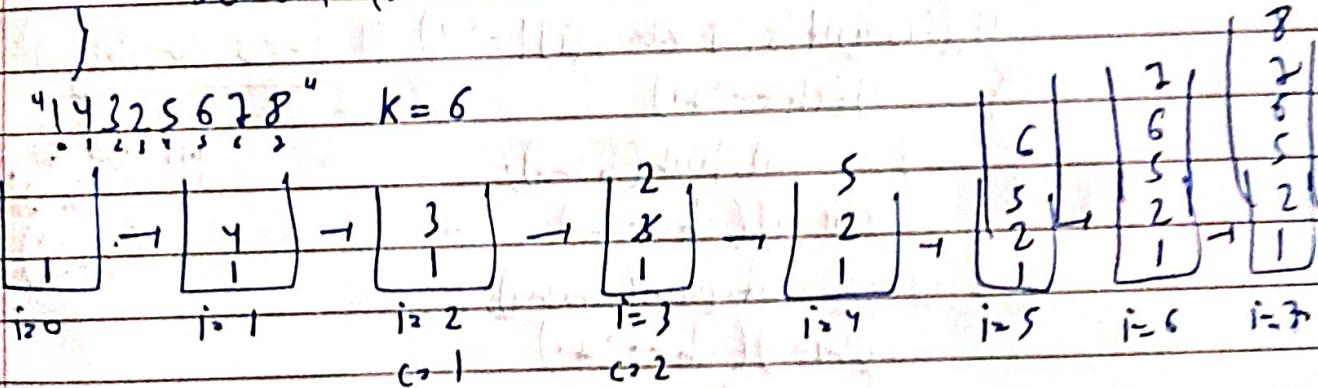
{
 if(s[i] == '0')
 s.pop_back();
 else
 break;
}

reverse(s.begin(), s.end());

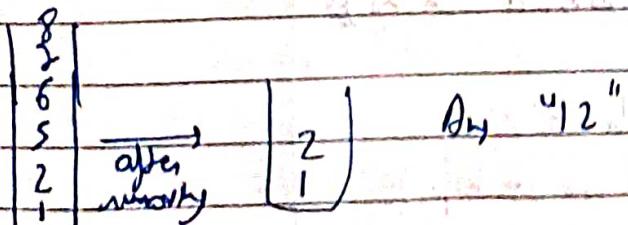
return (s.size() == 0) ? "0" : s; // final s = 2014

}

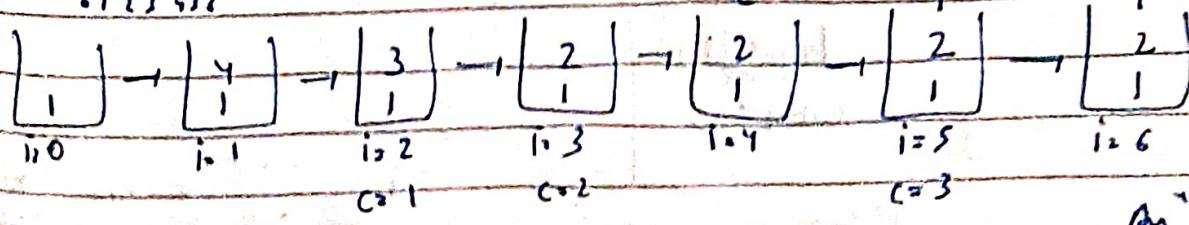
e.g. "14325678" k=6



$c=2$ & all 2 element of front & shift it to \downarrow , \therefore to get smallest no. we have to remove 4 element from last of stack.



e.g. "1432219" k=3



Ans: "1219"
Jahnu

(XII)

Basic Calculator II

- Given string evaluate it
- "123+2+2" | s can be ('+', '+', '+', '+')

Q122

```
int l(string s)
{ stack<int> st;
```

```
int num=0;
char ch='+';
```

```
for(int i=0; i<s.length(); i++)
```

```
y(!isdigit(s[i]))
```

```
num = (num*10) + (s[i] - '0');
```

```
y((!isdigit(s[i]) && s[i] != '+') || i==s.length()-1)
```

```
y(ch=='+')
```

```
st.push(num);
```

```
ch = y(ch=='-')
```

```
st.push(-num);
```

```
ch = y(ch=='*')
```

```
int a = st.top();
```

```
st.pop();
```

```
st.push(a*(num));
```

```
ch = y(ch=='/')
```

```
int a = st.top();
```

```
st.pop();
```

```
st.push(a/num);
```

```
ch = s[i];
```

```
num = 0;
```

'(' encountered $\rightarrow n+1$

so on

')] closed and $n \leftarrow 3$
 $\rightarrow \text{sign} = -1 \leftarrow \text{sum} = 12 + (-3)$
 $\text{sum} = -\text{sum} = -9 \leftarrow \text{sum} = 9$
 $\text{sum} = 9 + (\text{st}.top(), -9 + 7) \leftarrow -2$

(`isdigit()`): checks if given character is a digit or not

classmate

Digital
Power

```
int ans=0;  
while (!st.empty())  
{  
    ans+=st.top();  
    st.pop();  
}
```

$$\begin{array}{r} \cancel{S+3+6-S+7/1} \\ S+3 \\ \boxed{S} \end{array}$$

when one

$$\text{logic } S + 3 \neq 6 - S + 7 / 1$$

$$\underline{1} \quad \underline{s+18} - \underline{s+7}$$

$$(5) + (18) + (-5) + (2)$$

$(5) + (18) + (-7) + (7)$ \rightarrow यहाँ 21 से 7 घटा जाएगा (add sub ans)

Tonight they

କୁତୁହା ଦିନ ୧୯୮୧

XII Basic Calculations B

s (omit of '+' '-' '(' ')')

- unary operator $'-'$ can be used as unary operation like $"-1"$ & $"-(2+3)"$

opp - 5

0/b 12

- No two consecutive operators in the input.

$$\log_2(3+4-(1+(4+5+2)-3)+(6+8))$$

- sign of previous sign still 2π maintain after "3+4"
 $2\pi + \pi \neq \pi$ sign = + $\frac{1}{2}\pi$ (mean sign = 2)

\rightarrow sign \neq \rightarrow ('encountered
sign, -1 now starts)

-1

Wm 20
Wm 20
Sign 1

1
-
2

$$\leftarrow \sin_{\varepsilon-1}$$

$$\begin{bmatrix} -1 \\ 2 \end{bmatrix} \leftarrow \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

13' en
Signs 1
Sum 11

$$\frac{1}{\sin \theta} = 1$$

July 10

$$\text{sum} = 11 + 5 \cdot \text{dop}() = 11 + 1 = 12$$

TC O(n) SC O(n)

classmate

Date _____
Page _____

int f(string s)

{ stack<int> st;

long long int sign = 1, sum = 0, num = 0;

for (int i = 0; i < s.length(); i++)

} if (isdigit(s[i]))

num = 0;

while (i < s.length() && isdigit(s[i]))

{ num = (num * 10) + (s[i] - '0');

i++;

}

num = sign == -1 ? -1 * num : num;

sum += num;

}

} if (s[i] == '-')

sign = -1;

} if (s[i] == '+')

sign = 1;

} if (s[i] == '(')

st.push(num);

st.push(sign);

sum = 0;

num = 0;

sign = 1;

}

} if (s[i] == ')')

{ sign = st.top() == -1 ? -1 : 1;

st.pop();

sum = sign == -1 ? -1 * sum : sum;

sum += st.top();

st.pop();

}

} return sum;

IV Remove Duplicate letters

$s = "cbacdcbc"$ add remove duplicate characters so that every letter appears once & only once. You must make sure that your result is lexicographical smallest order.

string f(string s)

{ stacks char st;

vector<bool> v(26, 0); // visited vector bool

unordered_map<char, int> m; // frequency count

for(int i=0; i < s.length(); i++)

 m[s[i]]++;

 for(int i=0; i < s.length(); i++)

 while(!st.empty() && st.top() > s[i] && m[st.top()] > 0
 && v[s[i] - 'a'] == 0)

 v[st.top() - 'a'] = 0;

s[i] is current

}

 if(v[s[i] - 'a'] == 0) // if value == 0 means we can pick

 st.push(s[i]);

 v[s[i] - 'a'] = 1; // marking it as visited.

 m[s[i]]--;

}

string sl;

while(!st.empty())

 sl += st.top();

 st.pop();

reverse(sl.begin(), sl.end());

return sl;

e.g. "cbacdcbc"

$\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$

initially

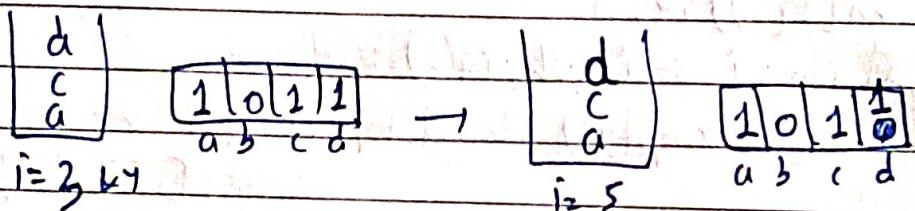
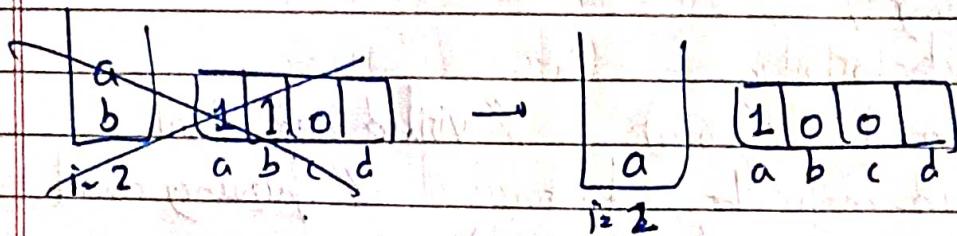
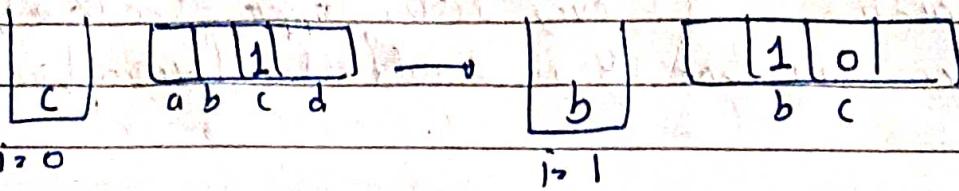
$$a = x_0$$

$$c = \text{XOR } x_0$$

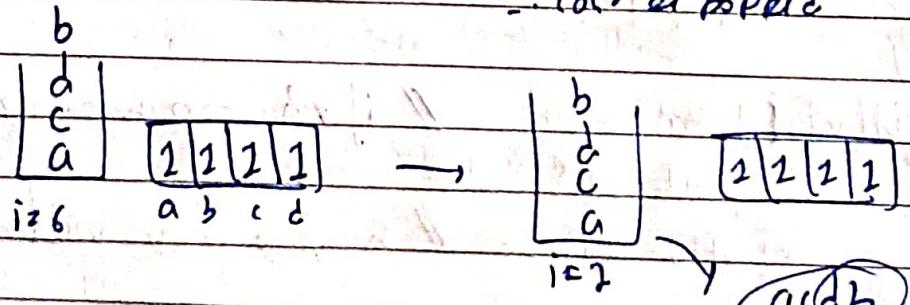
$$b = x_0$$

$$d = x_0$$

0	1	2	3	-
a	b	c	d	



Since $d = 0$
 $\therefore \text{can't be bubbled}$



\rightarrow **(acdb) Any**

e.g. "abacab"

$\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$

$$a = x_0$$

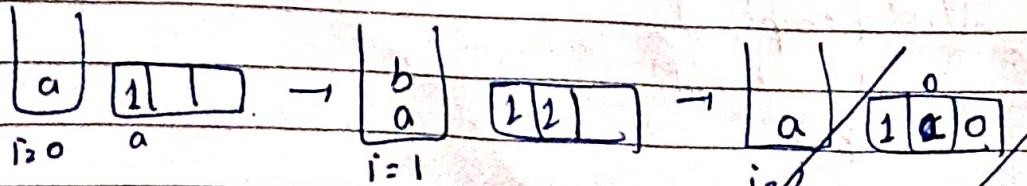
$$b = x_0$$

$$c = x_0$$

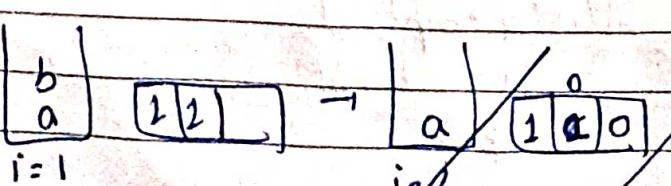
$$a = 2x_0$$

$$b = 2x_0$$

$$c = 1x_0$$

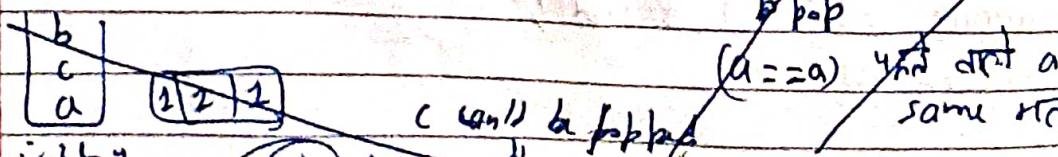


\rightarrow

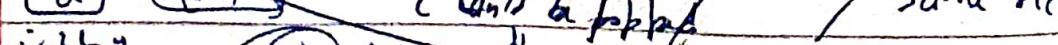


b.pop

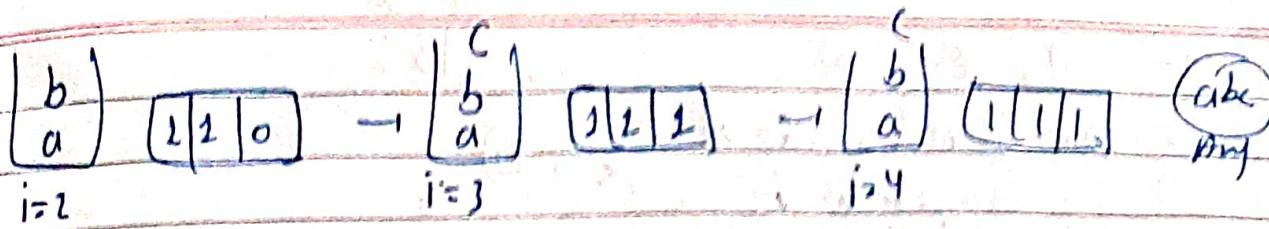
$(a == a)$ \therefore not a nt so bad
 same result (g/c)



c.pop b.pop



(abca) Any



⑩ 132 Pattern

132 pattern is a subsequence of three indices $i < j < k$ such that $i < j < k$ and $\text{num}[i] < \text{num}[k] < \text{num}[j]$ ($\text{num}[i] < \text{num}[k] < \text{num}[j]$)

- Naive: $O(n^3)$
- Brute Optimized: $O(n^2)$

```
bool f(vector<int> v)
{
    int mini = v[0];
    for(int j=0; j<v.size(); j++)
    {
        for(int k=j+1; k<v.size(); k++)
        {
            if(v[k] > mini && v[j] > v[k])
                return true;
            mini = min(mini, v[j]);
        }
    }
    return false;
}
```

g1	3 1 4 2	[-1, 3, 2, 0]
	132 exist [1, 4, 2]	[-1, 3, 2], [1, 3, 0], [1, 2, 0] fails

i j k

132

O(n) S - O(n)

```
bool f(vector<int> v)
{
    stack<int> st;
    int thidnumber = INT_MIN;
    for(int i = v.size() - 1; i >= 0; i--)

```

$v[i] < \text{thidnumber}$ when 1.

while(!st.empty() && st.top() < v[i])

$\text{thidnumber} = st.top();$

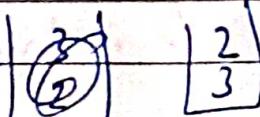
st.pop();

st.push(v[i]);

when 0:

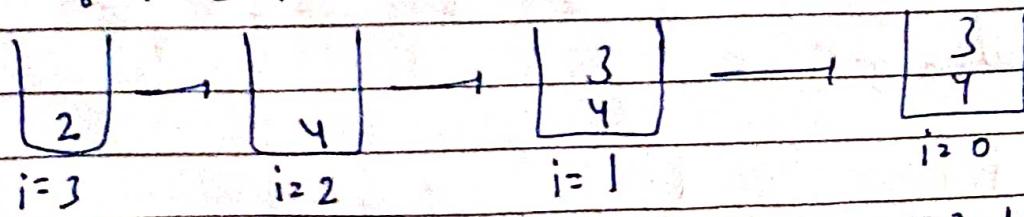
logik: $v[i] < v[k] < v[j]$

132 pattern: in stack we store 32 bottom



$v[j] \quad v[k]$

e.g. 1 3 4 2



$2 < 4$
 $th = 2$

$v[i] < th$

$3 < 2$

X

$v[i] < th$

$1 < 2 \sim \text{when } 1$

~~Q17~~ Sum of SubArray Minimum

Given array, find $\min(b)$ where b ranges over every subarray of a . answer may be large do $1 \cdot 10^9 + 7$

e.g. $[3, 1, 2, 4]$

all $b > 1$ subarray: $[3], [1], [2], [4], [3, 1], [1, 2], [2, 4], [3, 1, 2], [1, 2, 4], [3, 1, 2, 4] = 3 + 1 + 2 + 4 + 1 + 1 + 2 + 1 + 1 = 17$

int f(vector<int> v)

{ int n = v.size(), m = 1000000007;

stack<int> st;

int ans = 0;

for (int i = 0; i < n; i++)

 while (!st.empty() && (i == n || v[st.top()] >= v[i]))

 int t = st.top();

 st.pop();

 int l = st.empty() ? t + 1 : t - st.top();

 int r = st.empty() ? 1 - t : i - t;

 long long bmod = (l + 1) * m;

 bmod = (bmod + v[t]) * m;

 ans += (ans + bmod) / m;

 st.push(i);

return ans;

$v[t] + l + 1$

total no. of subarray with a given min t

logic 3 1 2 5 4 9 2 6 8
 4 के तक min 5 11, उस तक पहले 16 1
 से पहले होता element at mil 16 1

पहली i=5 91 +
 calculate

Date _____
 Page _____

i=1 3 1 2 5 4 9 2 6 8
 . 1 2 3 4 5 6 2 8

i=0

0

$$i=1 \quad \begin{array}{|c|} \hline 0 \\ \hline \end{array} \quad l = k+1 = 0+1=1 \quad l+1 \times v[0] = 1+1 \times 3 = 3 \\ n = i-k = 1-0=1 \quad \text{ans} = 3 \quad | \quad |$$

3>1

i=2,3

3
2
1

$$i=4 \quad \begin{array}{|c|} \hline 2 \\ \hline 1 \\ \hline \end{array} \quad l \rightarrow l++ = l = 1 - 1. \text{top}(1) = 3-2=1 \quad l \times v = 1 \times 5 \quad | \quad | \\ n = i-k = 4-3=1 \quad \text{ans} = 8$$

index

i=5

4
2
1

$$i=6 \quad \begin{array}{|c|} \hline 4 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} \quad l \rightarrow l++ = l = 5-4=1 \quad l \times v = 9 \quad | \quad | \\ n = 6-5=1 \quad \text{ans} = 12$$

$$i=7 \quad \begin{array}{|c|} \hline 4 \\ \hline 2 \\ \hline 1 \\ \hline 8 \\ \hline \end{array} \quad l \rightarrow l++ = l = 6-4=2 \quad l \times v = 14 \quad | \quad | \\ n = 7-6=1 \quad \text{ans} = 14$$

i=8

4
2
1

 inwards stack तर जाए
 मत (i=-n) 91 उत्तम

stack से वह index जोके right हो वह होता element हो
 $v[\text{st. top}()] \neq 1$

$i=9$ while loop condition via odd st $\text{push}()$ -> if true

$$\boxed{1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9} \quad l = 8 + 1 = 9 - 7 = 2 \quad \text{div} = 8 \\ n = 9 - 8 = 1 \quad \text{ans} = 31 + 8 = 39$$

$$\boxed{1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9} \quad l = 7 - 4 = 3 \quad \text{div} = 6 + 6 = 36 \\ n = 9 - 7 = 2 \quad \text{ans} = 39 + 36 = 75$$

$$\boxed{1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9} \quad 3 \ 1 \ 2 \ 5 \ 4 \ \underline{\underline{9 \ 2 \ 6 \ 8}} \quad \left. \begin{array}{l} \text{6 path} \\ 6 \times 6 = 36 \end{array} \right\}$$

$$\boxed{1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9} \quad l = 4 - 2 = 2 \quad \text{div} = 10 + 4 = 40 \\ n = 9 - 4 = 5 \quad \text{ans} = 115$$

$$3 \ 1 \ 2 \ 5 \ 4 \ \underline{\underline{9 \ 2 \ 6 \ 8}} \quad \left. \begin{array}{l} \text{6 path} \\ 6 \times 6 = 36 \end{array} \right\}$$

$$\boxed{1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9} \quad l = 2 - 1 = 1 \quad \text{div} = 2 \times 2 = 14 \\ n = 9 - 2 = 7 \quad \text{ans} = 129$$

$$3 \ 1 \ 2 \ 5 \ 4 \ \underline{\underline{9 \ 2 \ 6 \ 8}} \quad \left. \begin{array}{l} \text{2 path} \\ 2 \times 2 = 4 \end{array} \right\}$$

$$\boxed{1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9} \quad l = 1 + 1 = 2 \quad \text{div} = 16 + 1 = 16 \\ n = 9 - 1 = 8 \quad \text{ans} = 145$$

$$3 \ 1 \ 2 \ 5 \ 4 \ \underline{\underline{9 \ 2 \ 6 \ 8}} \quad \left. \begin{array}{l} \text{16 path} \\ 16 \times 16 = 256 \end{array} \right\}$$

Note

PLF → for loop traversal (i=0 to n-1)

NLG ← (i=n-1 to 0)

classmate

Date _____
Page _____

(XVII)

Sum of Subarray Range:

The range of subarray of array is the difference between largest element & smallest element in the subarray.

- Return sum of all subarray ranges.

e.g: [1, 2, 3]

{1}, {2}, {3} → sum 0

[1, 2] = 1, [2, 3] = 3 - 2 = 1, [1, 2, 3] = 3 - 1 = 2 sum 2 + 0 + 1 + 1 = 4

long long f(v<vector> v)

{ stack<int> st;

long long res=0, l, r;

int n=v.size();

for(int i=0; i<n; i++)

min element
of every subarray

while(!st.empty() && (i==n || v[st.top()]>=v[i]))

{ int t=st.top();

st.pop();

l=st.empty() ? i+1 : st.top();

r=i-t;

res+= (long long) (v[t]*l+r);

st.push(i);

st=stack<int>(); (initialization)

for(int i=0; i<n; i++)

{ while(!st.empty() && (i==n || v[st.top()]<=v[i]))

{ int t=st.top();

st.pop();

~~and~~ Clement
~~(N.Y.)~~ N.B. Augr

```

a) l = st.empty() ? t+1 : t-st.top();
n = i-t;
m += (long long)(v[t]*l+n);
}
st.push(i);
}
st.push(i);
}
when m;

```

logic: first subarray array में से min element निकाला जाए
~~सब~~ ~~लॉप~~ के द्वारा and max element निकाला जाए
substract कर दिया

(XVII) Shortest Unsorted Continuous Subarray :

- You have to find CS that if you sort this subarray in ascending order, whole array is sorted.
 - when length of subArray

`int f(vector<int> v)`

{ int n=v.size(); minz=INT_MAX , maxi=INT_MIN;

$y/\sin x = 0$ when $x = 0$

```
for(int i=0 ; i< v.size(); i++)
```

$y(i=0)$

$\leftarrow \text{if } (\text{v}[i] > \text{v}[i+1])$

$$\min_i = \min(\min_i, v[i]);$$

maxi = max(maxi, v[i]);

it pattern
विकास

$(\forall v[i] \leq v[i+1]) \quad || \quad (\forall v[i] \geq v[i-1])$
i) ना तरीका problem

classmate
Date _____
Page _____

else if ($i == n - 1$)

{
 } ($v[i] < v[i-1]$)

 mini = min(mini, v[i]);

 maxi = max(maxi, v[i]);

}

} else if ($v[i] < v[i-1] \quad || \quad v[i] > v[i+1]$)

{
 mini = min(mini, v[i]);

 maxi = max(maxi, v[i]);

 → all this branch
 condition (loop by //
 written)

if (maxi == INT_MIN || mini == INT_MAX) return 0;
int i, j;

for (i = 0; i < n && v[i] <= mini; i++); // finding mini pos
for (j = n - 1; j >= 0 && v[j] >= maxi; j--); // — maxi —

when $j = i + 1$;

logic: mini और min element दोनों वह उत्तरांश हैं, फिर
check करें कि mini कहाँ पर आया।

• maxi और max element दोनों वह उत्तरांश हैं, फिर check
करें कि maxi कहाँ पर आया।

e.g. [2, 6, 4, 8, 10, 9, 15]

mini = 4

maxi = 15 & 9

→ now find $\rightarrow j \neq (j-i+1)$

position

Maximum Width Ramp

- A ramp in an array is a pair (i, j) for which $i < j$ & $\text{nums}[i] \leq \text{nums}[j]$. The width is $(j - i)$.
- Within max width ramp.

e.g. $[6, 0, 8, 2, 1, 5]$
 $\Rightarrow b = 4$

```

int f(vector<int> v)
{
    stack<int> st;
    int n = v.size(), ans = 0;
    for (int i = 0; i < n; i++)
    {
        while (st.empty() || v[i] < v[st.top()])
            st.push(i);
    }
    for (int i = n - 1; i >= 0; i--)
    {
        while (st.empty() || v[st.top()] <= v[i])
            ans = max(ans, i - st.top());
        st.pop();
    }
    return ans;
}
    
```

1
2
4
6

logic: ~~first~~ for loop it is clearly stack ~~at~~ ~~in~~ then iterate from last.

e.g. $[6, 1, 5, 4, 3, 2, 1, 4]$

1	1
0	6

now iterate from last $\Leftarrow 4 > v[\text{st.top}]$ true
 $\Rightarrow \text{ans} = 6$ (by 4th)

st.pop() 0

$\Leftrightarrow 4 = 6 \times \left\lfloor \frac{1}{2} \right\rfloor + 1$ $\Rightarrow 6 \times \left\lfloor \frac{1}{2} \right\rfloor - 1$
 $i--$ $i--$ no ans possible $\therefore \text{ans} = 5$

→ NGF

$O(N) \quad O(N)$

$[1, 3, 2, 4]$

$[6, 8, 0, 1, 3]$

O/p $[3, 4, 4, -1]$

O/p $[8, -1, 1, 3, -1]$

`vector<int> f(vector<int> a, int n)`

`vector<int> ans(n, -1);`

`stack<int> st;`

`for(int i = n-1; i >= 0; i--)`

`while(!st.empty() && st.top() <= a[i])`

`st.pop();`

`ans[i] = st.empty() ? -1 : st.top();`

`st.push(a[i]);`

`}`

`return ans;`

`}`