Postfix expression calculation

Str = 4 5 7 2 + − *

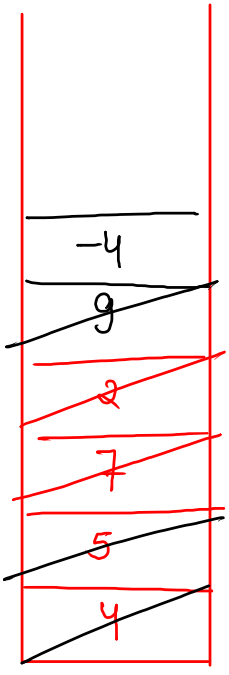
5 9

+ ⇒ val2 + val1

− ⇒ val2 − val1

* ⇒ val2 * val1

/ ⇒ val2 / val1



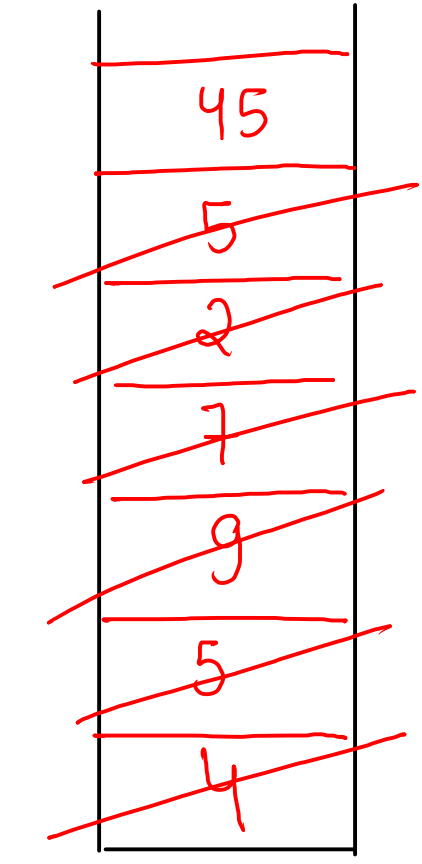
Stack
<Integer>

val1 = ~~2~~ ~~9~~ -4

val2 = ~~7~~ ~~5~~ 4

ans = ~~9~~ ~~-4~~ -16

str = 45+72-*



Stack

psudo code:-

if ch is number
then push in st

else
pop 2 values
and solve.
(and push ans again)

val1 = ~~5~~ ~~2~~ 5

val2 = ~~4~~ ~~7~~ 9

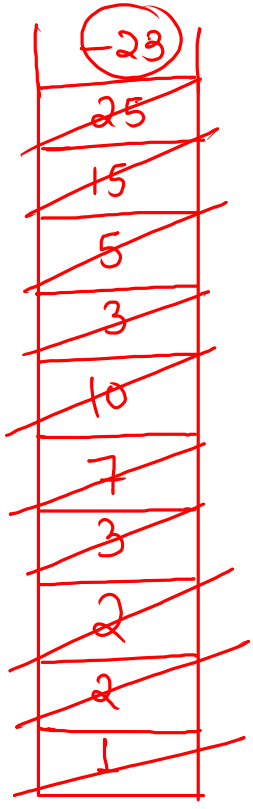
ans = ~~9~~ ~~5~~ 45 ✓

code

```java
public static void postfixExp(String str) {
    Stack<Integer> st = new Stack<Integer>();
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        if ( Character.isDigit(ch) ) {
            st.push( ch - '0' );
        } else {
            int val1 = st.pop();
            int val2 = st.pop();
            int ans = -1;
            if (ch == '+') {
                ans = val2 + val1;
            } else if (ch == '-') {
                ans = val2 - val1;
            } else if (ch == '*') {
                ans = val2 * val1;
            } else if (ch == '/') {
                ans = val2 / val1;
            }
            st.push(ans);
        }
    }
    System.out.println( st.pop() );
}
```

dry run

12 * 37 + 35 * + −



val1 = ~~2~~ ~~7~~ ~~5~~ ~~15~~ 25

val2 = ~~1~~ ~~3~~ ~~3~~ ~~10~~ 2

ans = ~~2~~ ~~10~~ ~~15~~ ~~25~~ -23

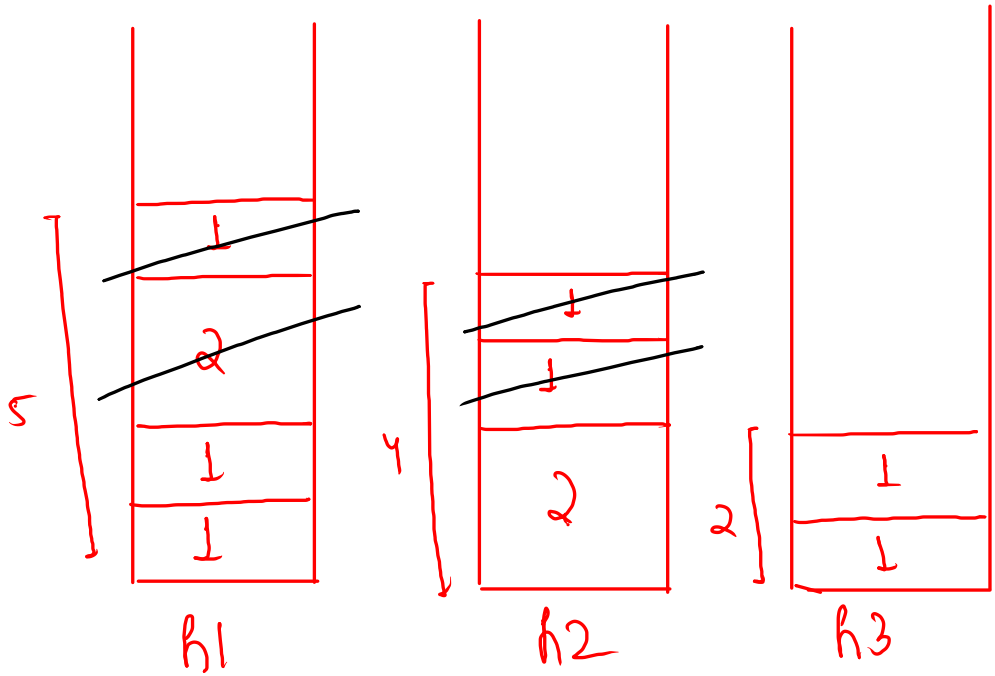
ans = -23

Equal Stacks

$$h1 = [1, 2, 1, 1]$$
$$h2 = [1, 1, 2]$$
$$h3 = [1, 1]$$

Approch :-

- remove top element from stack which is of maximum height
- until all stack have equal height



h = 2

Next greater element on left

arr

3	4	3	2	1

ans

-1	-1	4	3	2

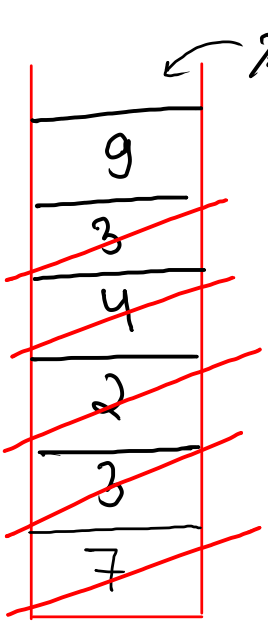
---

arr = 7 6 5 9 3 8 4

ans = -1 7 6 -1 9 9 8

arr

7	3	2	4	3	9



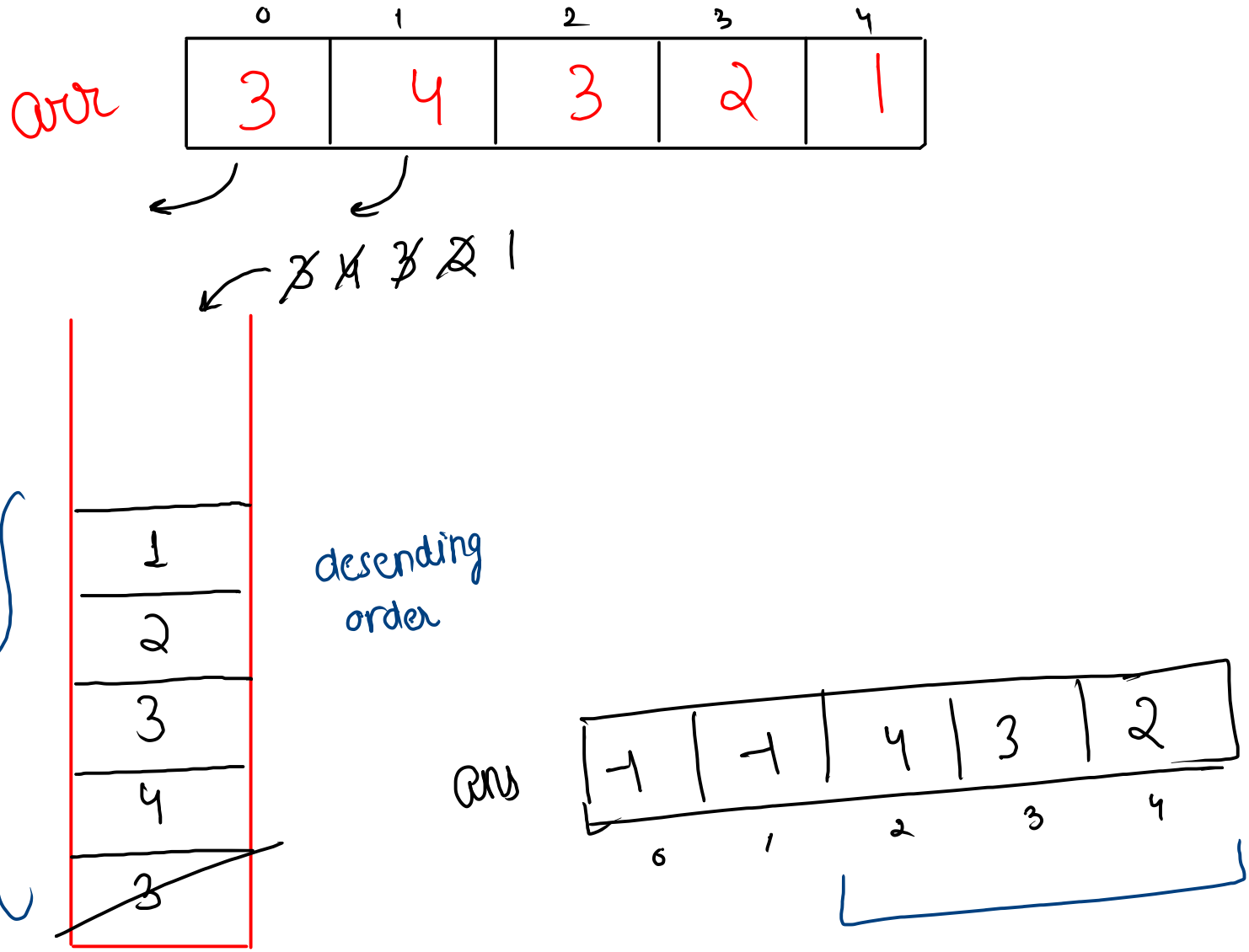
~~7~~ ~~4~~ ~~3~~ 9

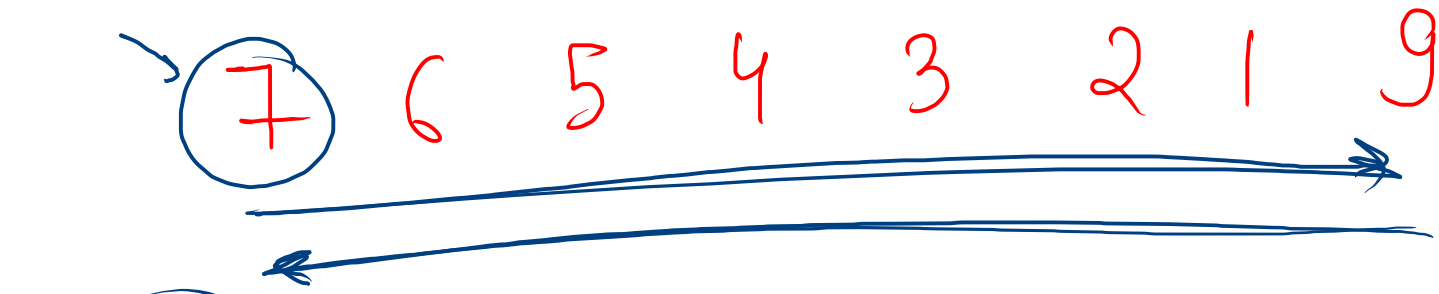
approch :- I will try to create a stack in desencding order

Note :- if top element is greater than curr. then top element is ans for curr.
else, that is not an ans for me or any other no. in front of me

ans

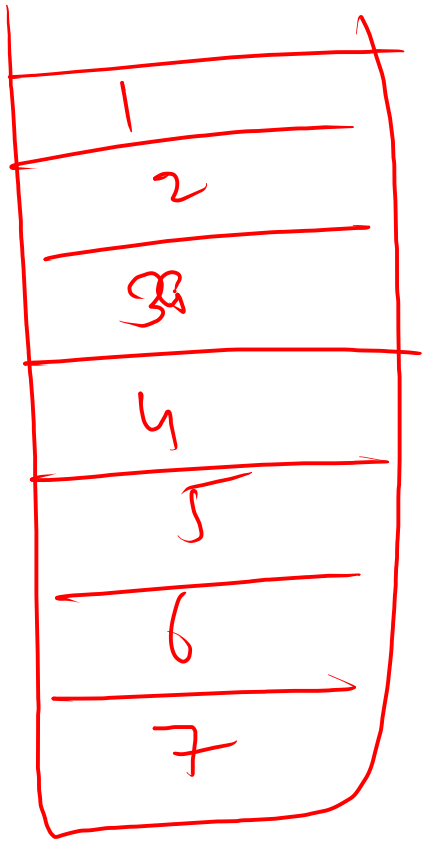
-1	7	3	7	4	-1


0
1
2
3
4
arr
3
4
3
2
1
1
2
3
4
3
descending order
ans
-1
-1
4
3
2
0
1
2
3
4



2 x N

N



O(N*N)

O(N+N)

T.C = O(N)

H.W

- Greater element on left
- Greater element on right
- Smaller element on left
- Smaller element on right

Code

```java
public static void nextGreaterOnLeft(int[] arr) {
    int[] ans = new int[arr.length];

    Stack<Integer> st = new Stack<Integer>();
    for (int i = 0; i < arr.length; i++) {
        while ( !st.isEmpty() && st.peek() <= arr[i] ) {
            st.pop();
        }
        if (!st.isEmpty()) {
            ans[i] = st.peek();
        } else {
            ans[i] = -1;
        }
        st.push( arr[i] );
    }

    for (int i : ans) {
        System.out.print(i + " ");
    }
}
```