Stack

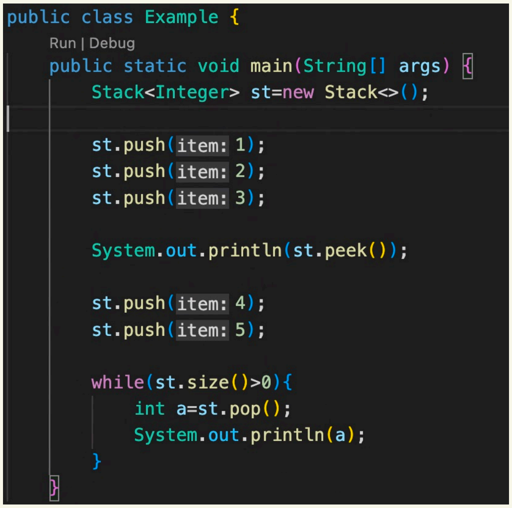
LIFO → last in, first out
semantics

⇒ Syntax

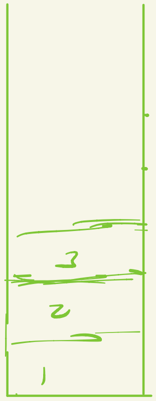
Stack < Integer> st = new Stack<>();

⇒ function

1) add   st. push (5)
2) remove) int a= st. pop( ) → remove and return the top element
3) int a= st. peek(); → returns the top element
4) size) st. size ();



```
public class Example {
    Run | Debug
    public static void main(String[] args) {
        Stack<Integer> st=new Stack<>();

        st.push(item: 1);
        st.push(item: 2);
        st.push(item: 3);

        System.out.println(st.peek());

        st.push(item: 4);
        st.push(item: 5);

        while(st.size()>0){
            int a=st.pop();
            System.out.println(a);
        }
    }
```

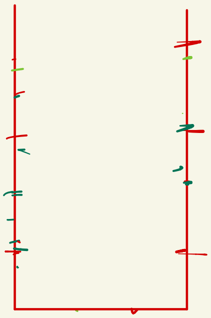


int a=st.pop()

5
4
↳ 3
2
1

{ [( ]) } ✗ → balanced

{ { { } } ( ) } ✓



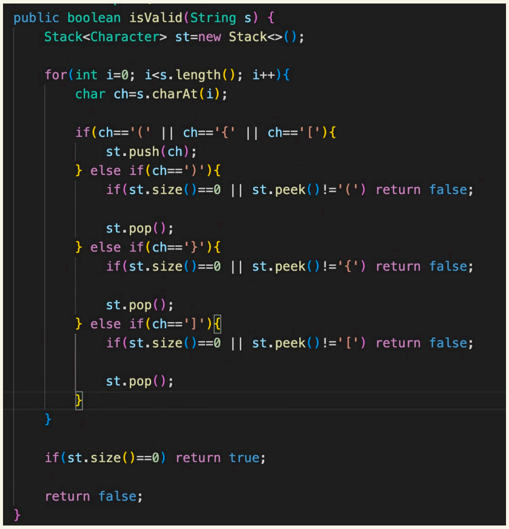
st.peek()

( ) { } [ ] ✓

((( )) { }

int a = ~~5~~ 7;

if (a != 5) return false



```
public boolean isValid(String s) {
    Stack<Character> st=new Stack<>();

    for(int i=0; i<s.length(); i++){
        char ch=s.charAt(i);

        if(ch=='(' || ch=='{' || ch=='['){
            st.push(ch);
        } else if(ch==')'){
            if(st.size()==0 || st.peek()!='(') return false;

            st.pop();
        } else if(ch=='}'){
            if(st.size()==0 || st.peek()!='{') return false;

            st.pop();
        } else if(ch==']'){
            if(st.size()==0 || st.peek()!='[') return false;

            st.pop();
        }
    }

    if(st.size()==0) return true;

    return false;
}
```

Ques input string ( ) ≠

a = ~~0~~ ~~1~~ ~~2~~ ~~1~~ 0

O(1) space
O(N) time

if (a < 0) return false

(( ) ( ))

(( )) (( ) ( ))

a = ~~0~~ ~~1~~ ~~2~~ ~~1~~ ~~0~~ ~~1~~ ~~2~~ ~~1~~ ~~2~~ ~~1~~ 0

( )) (

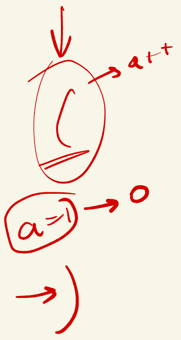
if (ch == '(') a++;
else a--;

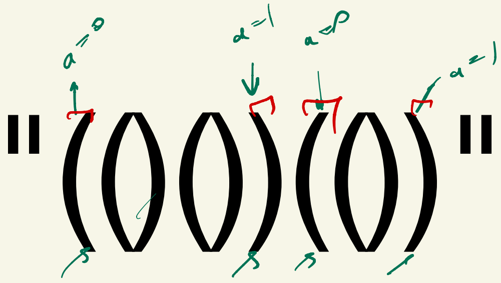
if (a == 0) return true

a = 1

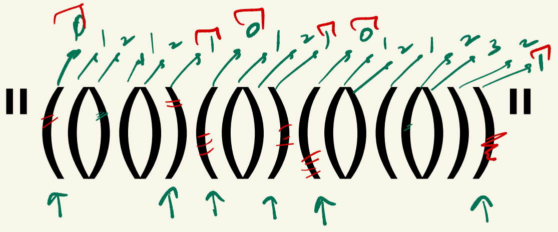
(( ) ( )

a = ~~1~~ ~~2~~ ~~1~~ ~~2~~ 1





a
↳ number of opening brackets with no corresponding closing bracket



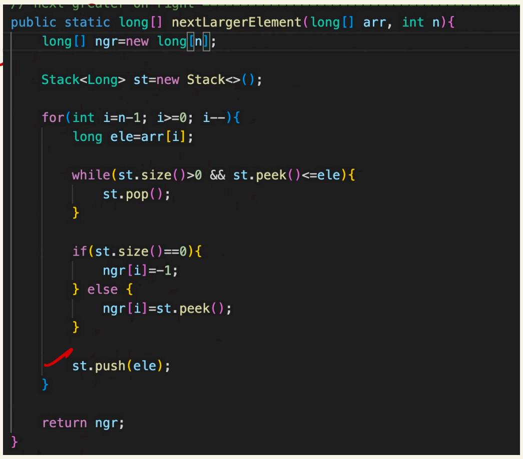
a++

if (ch=='(' and a==0)

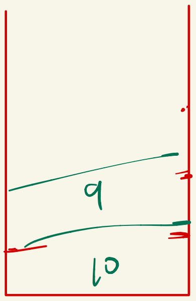
if (ch==')' and a==1)

Next greater

0	1	2	3	4	5	6	7	8	9	10
6	14	0	10	3	9	15	4	2	1	12
1	6	3	6	5	6	-1	10	10	10	-1



```
// next greater on right -----------------------------------------
public static long[] nextLargerElement(long[] arr, int n){
    long[] ngr=new long[n];

    Stack<Long> st=new Stack<>();

    for(int i=n-1; i>=0; i--){
        long ele=arr[i];

        while(st.size()>0 && st.peek()<=ele){
            st.pop();
        }

        if(st.size()==0){
            ngr[i]=-1;
        } else {
            ngr[i]=st.peek();
        }

        st.push(ele);
    }

    return ngr;
}
```



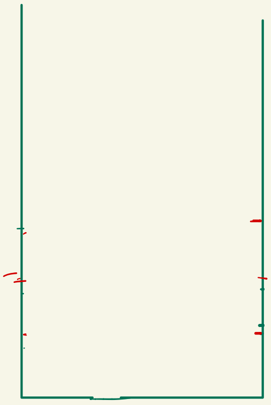
ngr(9)

arr(st.peek())

$$\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 8, & 5, & 3, & 4, & 3 \\ -1 & 6 & 4 & 6 & 6 \end{array}$$

$(2*n - 1 \to 0$

$9 \to 0$



$9 \% 5 \to 4$
$8 \% 5 \to 3$
$7 \% 5 \to 2$
$6 \% 5 \to 1$
$5 \% 5 \to 0$
$4 \% 5 \to 4$
$3 \% 5 \to 3$
$2 \% 5 \to 2$
$1 \% 5 \to 1$
$0 \% 5 = 0$

$$\begin{array}{cccccc} 7 & 3 & 1 & 5 & 2 & 3 \\ & \downarrow & & & & \\ -1 & 5 & 5 & 7 & 3 & 7 \end{array}$$

$$\begin{array}{ccccc} 2 & 1 & 5 & 4 & 3 \\ 5 & 5 & -1 & -1 & -1 \end{array}$$

---

0 → amar
1 → akbar
2 → anthony

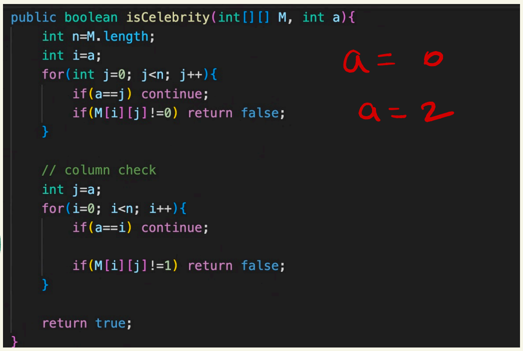
$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 \end{array}$$

$M(i)(j) = 0$
↓
i doesn't know jth person

$\boxed{a}$ row → everywhere 0 ~~→~~

a col → everywhere 1

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 1 & 0 \end{array}$$



```java
public boolean isCelebrity(int[][] M, int a){
    int n=M.length;
    int i=a;
    for(int j=0; j<n; j++){
        if(a==j) continue;
        if(M[i][j]!=0) return false;
    }

    // column check
    int j=a;
    for(i=0; i<n; i++){
        if(a==i) continue;

        if(M[i][j]!=1) return false;
    }

    return true;
}
```

a = 0

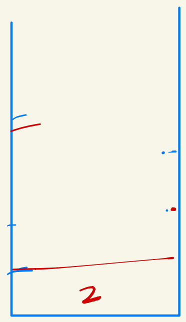
a = 2

Elimination method

3, 2
2, 1
2, 0

$\{$ M(a)(b) = 0
b can't be a celebrity

$\{$ M(a)(b) = 1
a can't be celebrity



tar = 2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0,	0,	1,	1,	1,	2,	2,	2,	2,	2,	2,	2,	2,	3,	3,	4,

$\uparrow$ si $\qquad\qquad$ ei

fo = ~~-1~~ 7

lo = 7

mid = $\frac{(0+15)}{2} \to 7$

ei = mid - 1

```
if (arr(mid) == tar)
    fo = mid
    right = mid - 1
if (arr(mid) == tar)
    lo = mid
    left = mid + 1
```