Reverse Words in a Given String

str = "reverse_words_in_a_string";

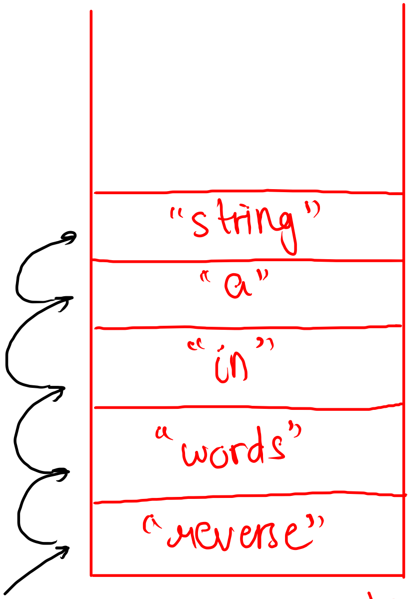
0 1 2 3 4 5 6 7 8 . . .

Inbuilt f^n

String[] arr = str.split(" ");

arr =

reverse,	words,	in,	a,	string
0	1	2	3	4



stack of string type

ans = ans + peek or peek + ans

split function

str = "reverse_words_in_a_string";

0 1 2 3 4 5 6 7 8 . . . .

str.split("e");

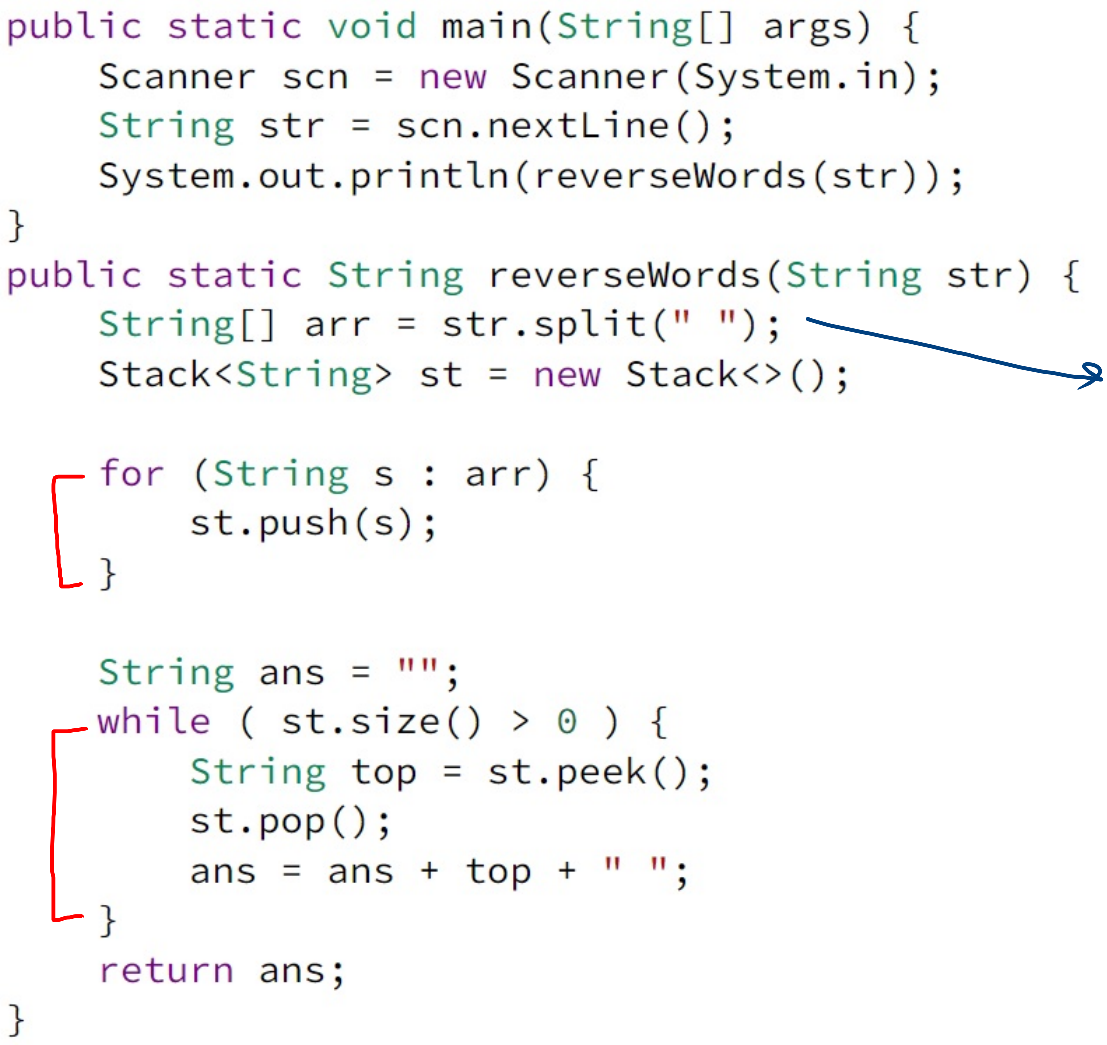
arr =

"r"	"v"	"rs"	"_words_in_a_string"

0

str.split("ord");

reverse_w	s_in_a_string



```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    System.out.println(reverseWords(str));
}
public static String reverseWords(String str) {
    String[] arr = str.split(" ");
    Stack<String> st = new Stack<>();

    for (String s : arr) {
        st.push(s);
    }

    String ans = "";
    while ( st.size() > 0 ) {
        String top = st.peek();
        st.pop();
        ans = ans + top + " ";
    }
    return ans;
}
```

T.C = $O(n)$

string type of arr

S.C = $O(n)$

n = stack size

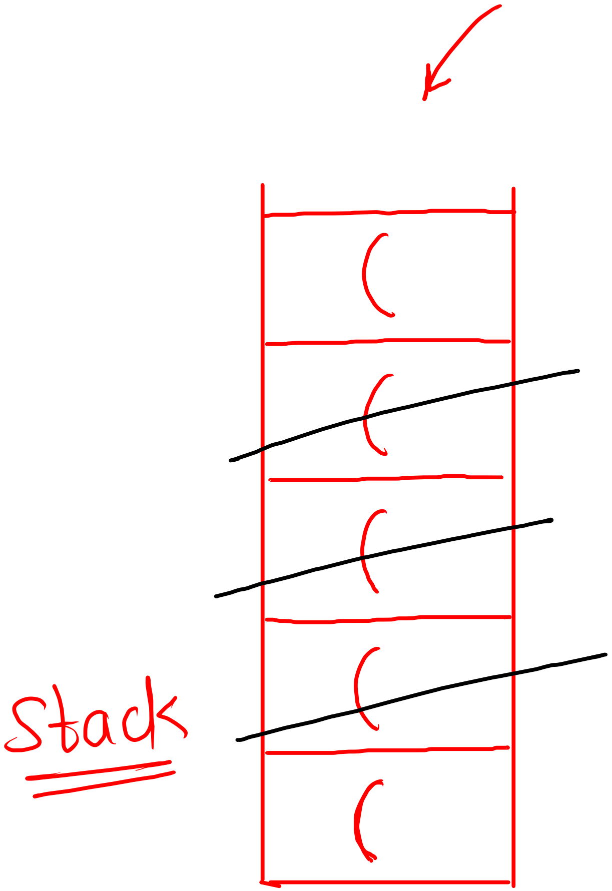
Longest Valid Parentheses 4

str = "(()(())("

0 1 2 3 4 5 6 7

ans = str.len - st.size

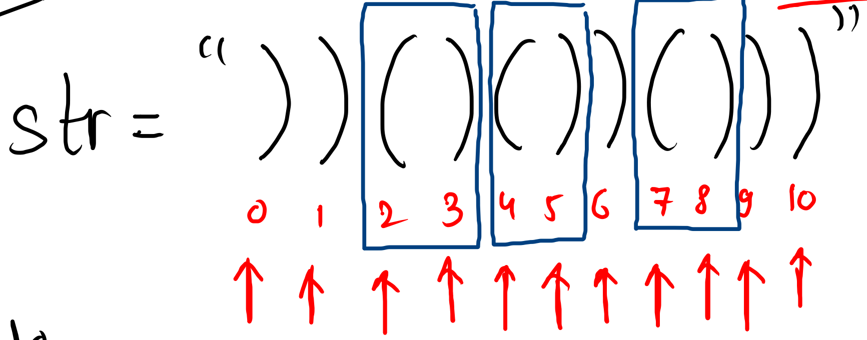
ans = 6

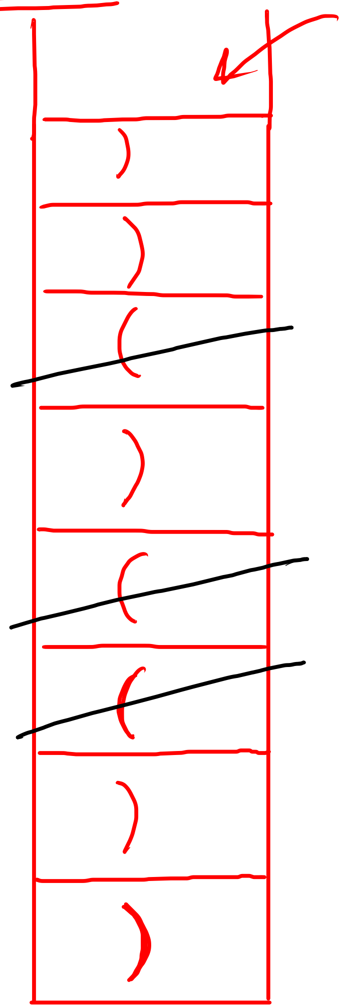


faith :- we always keep invalid para. in stack

Ex :-

ans = 11 - 5 = 6





size = 5

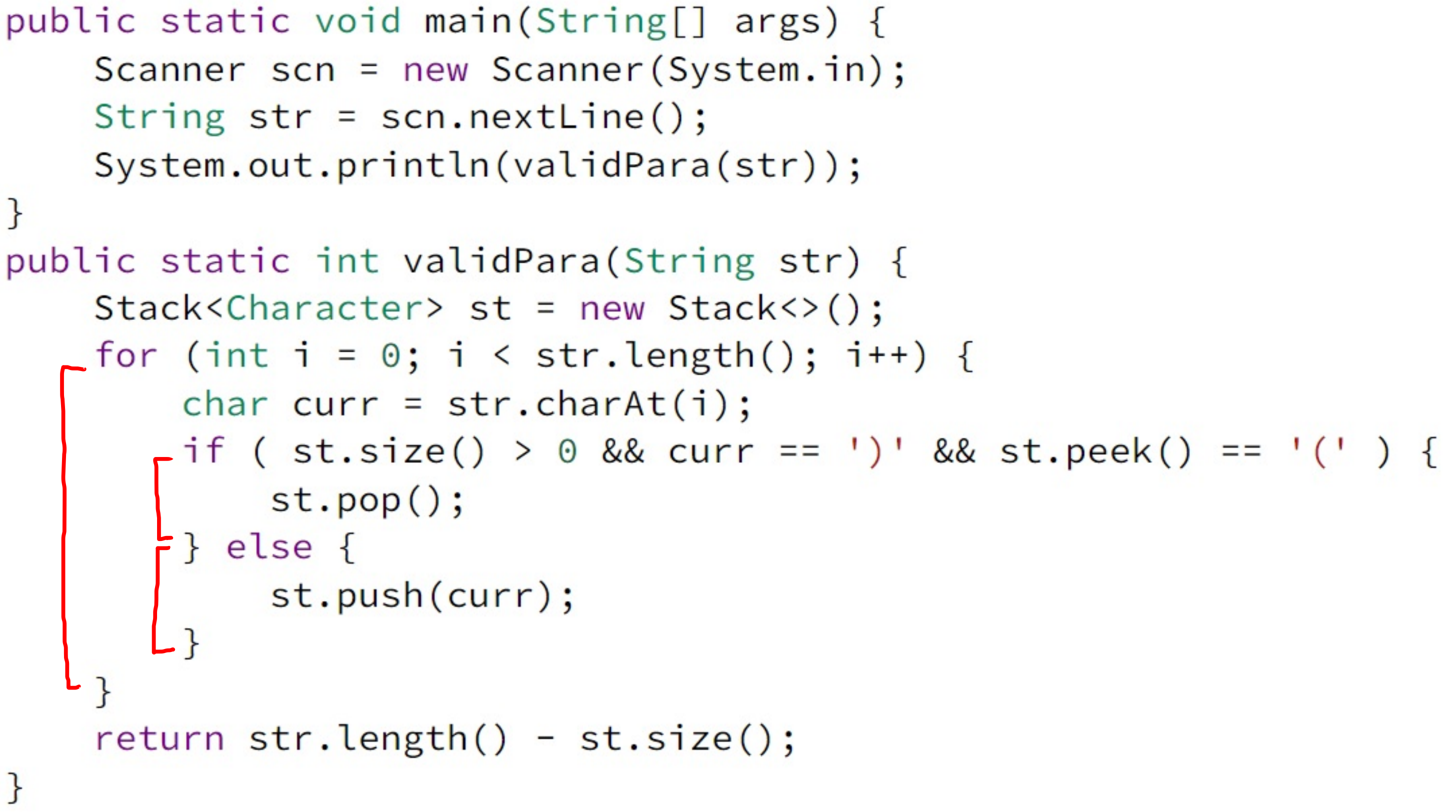
psudo code

```
1) traverse in string
    1.1) if curr ele == ) and top == (
            pop
    1.2) else
            push
```

Code

$T.C = O(n)$

$S.C = O(n)$



```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    System.out.println(validPara(str));
}
public static int validPara(String str) {
    Stack<Character> st = new Stack<>();
    for (int i = 0; i < str.length(); i++) {
        char curr = str.charAt(i);
        if ( st.size() > 0 && curr == ')' && st.peek() == '(' ) {
            st.pop();
        } else {
            st.push(curr);
        }
    }
    return str.length() - st.size();
}
```

Postfix expression calculation

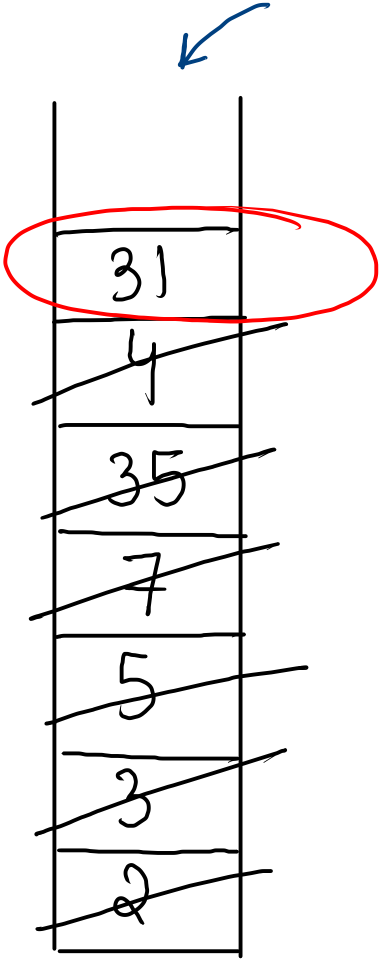
Prefix :- $- * + 2 3 7 4$

Infix :- $((2+3)*7)-4 = 31$

Postfix :- $\boxed{2 3 + 7 * 4 -} = 31$

str = "23+7*4-"

↑ ↑ ↑ ↑ ↑ ↑ ↑



ab+

↑ ↑ ↑

ans = 35 - 4

= 31

psudo code

```
1) traverse in string
    1.1) if num
            push

    1.2) else
            pop
            pop
            calculate (+, -, *, /)
            push ans in stack
```