

## Count Sort Algorithm

$$\min = -2$$

$$\max = 7$$

arr →	<table border="1"> <tr><td>-2</td><td>1</td><td>4</td><td>1</td><td>3</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> </table>	-2	1	4	1	3	0	1	2	3	4
-2	1	4	1	3							
0	1	2	3	4							

Sample Input 0

5

7 -2 4 1 3

Sample Output 0

-2 1 3 4 7

Step 01 → min, max element

Step 02 → Make use freq array whose size  $\Rightarrow \max - \min + 1$ 

Show the freq array.

freq of every character.

-2	-1	0	1	2	3	4	5	6	7
1	0	0	2	0	1	1	0	0	1
0	1	2	3	4	5	6	7	8	9

$$7 - (-2) + 1 = 10$$

$$\text{int pos} = \text{arr}[i] - \min$$

$$\text{freq}[\text{pos}]++;$$

Step 03 → Prefix sum of freq array.

-2	-1	0	1	2	3	4	5	6	7
1	1	1	2	3	2	4	4	4	5
0	1	2	3	4	5	6	7	8	9

$$7 - (-2) = 9$$

$$-2 - (-2) = 0$$

$$4 - (-2) = 6$$

$$2 - (-2) = 3$$

$$3 - (-2) = 5$$

Step 04 → filling the ans. →

ans	<table border="1"> <tr><td>-2</td><td>1</td><td>3</td><td>4</td><td>7</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> </table>	-2	1	3	4	7	0	1	2	3	4
-2	1	3	4	7							
0	1	2	3	4							

$$\text{int pos} = \text{freq}[\text{arr}[i] - \min] - 1$$

$$\text{pos} = 2$$

$$\text{ans}[\text{pos}] = \text{arr}[i]$$

$$[\text{freq}[\text{arr}[i] - \min] - 1]$$

$$i = 2$$

$$= \text{freq}[3 - (-2)] - 1$$

$$\text{freq}[5] - 1$$

$$\text{pos} = 2 - 1$$

$$i = 1$$

$$\text{freq}[7 - 2 - (-2)] - 1$$

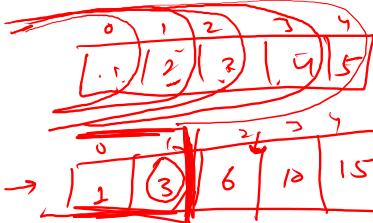
$$\text{freq}(8) - 1$$

$$\text{freq} = 0$$

for (int i = 0; i &lt; n; i++)

{}

$$\text{ans}[i] = \text{ans}[i];$$

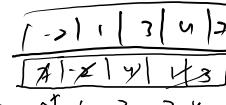


prefix sum array.



$$\text{prefix sum}(i) = \text{prefix sum}(i-1) + \text{arr}(i)$$

$$i = 0$$



$$\text{freq}[7 - 1 - (-2)] - 1$$

$$\text{freq}(9) - 1$$

$$5 - 1 = 4$$

$$i = 3$$

$$\text{freq}[1 - (-2)] - 1$$

$$\text{freq}[3] - 1$$

$$i = 2$$

$$\text{freq}[4 - (-2)] - 1$$

$$\text{freq}(6) - 1$$

$$4 - 1 = 3$$

```

class Solution {
    public static void countSort(int arr[], int n, int min, int max) {
        int freq[] = new int[max - min + 1];
        for (int i = 0; i < n; i++) {
            freq[arr[i] - min]++;
        }
        for (int i = 1; i < freq.length; i++) {
            freq[i] += freq[i - 1];
        }
        int ans[] = new int[n];
        for (int i = arr.length - 1; i >= 0; i--) {
            int pos = freq[arr[i] - min] - 1;
            ans[pos] = arr[i];
            freq[arr[i] - min]--;
        }
        for (int i = 0; i < n; i++) {
            arr[i] = ans[i];
        }
    }
}

```

T.C.  $\rightarrow O(\text{freq array}) \Rightarrow \cancel{\text{Linear}}$

S.C.  $\rightarrow O(\text{freq array})$

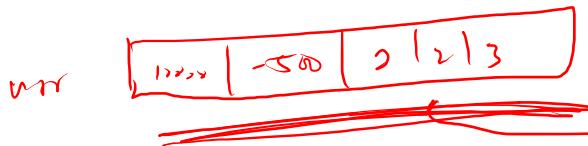
8067012

min & max.

max = 10000

min = -500

$$\text{freq} = 10000 - (-500) + 1$$



$\Rightarrow$  lot of space

1. You are given an array( $\text{arr}$ ) containing only 0's, 1's, and 2's.

2. You have to sort the given array in increasing order and in linear time.

#### Input Format

An Integer  $N$  arr1 arr $_2 \dots n$  integers

#### Constraints

- $1 \leq N \leq 10000$

- $\text{arr}[i] = 0, 1, 2$

#### Output Format

Sorted array

#### Sample Input 0

```
10  
1 0 2 2 1 0 2 1 0 2
```

#### Sample Output 0

```
0 0 0 1 1 1 2 2 2 2
```

Sort 0 1

X 2

0, 1, 2

0

while ( $i < n$ ) X  $\rightarrow (k - n + 1)$   
 $\curvearrowleft 2$



white ( $i \leq k$ )

if ( $\text{arr}[i] == 0$ )  
 $\text{swap}(i, j);$

$i++$   
 $j++;$

, else if ( $\text{arr}[i] == 1$ )  
 $i++;$

else {  
 $\text{swap}(i, k);$

$k--$

```

public static void swap(int arr[], int i, int j){
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

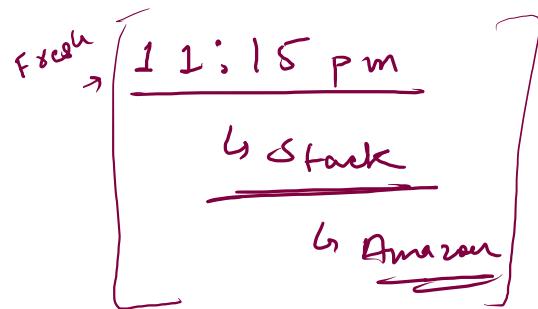
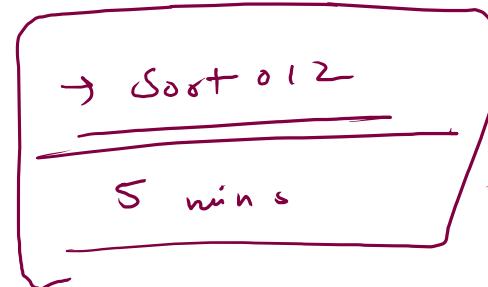
public static void sort012(int arr[], int n){
    int i=0, j=0, k=n-1;

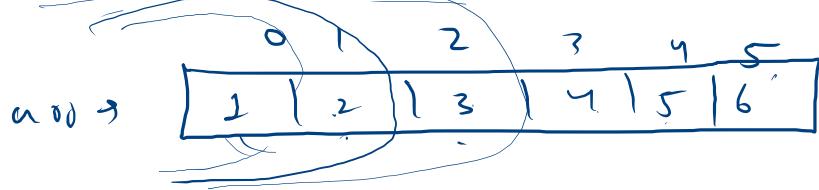
    while(i<=k){
        if(arr[i] == 0){
            swap(arr, i, j);
            i++;
            j++;
        }else if(arr[i] == 1){
            i++;
        }else {
            swap(arr, i, k);
            k--;
        }
    }
}

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int arr[] = new int[n];
    for(int i=0;i<n;i++){
        arr[i] = scn.nextInt();
    }

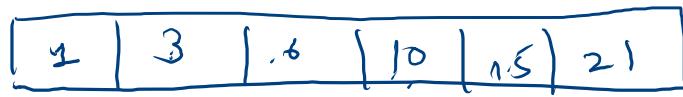
    sort012(arr,n);
    for(int i=0;i<n;i++){
        System.out.print(arr[i] + " ");
    }
}

```





prefix sum  
array



Stack -

↳ Linear D.S.

↳ LIFO (Last In First Out)

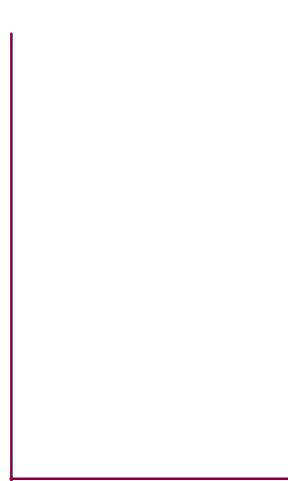
↳ Browser History

↳ Balancing brackets

(( { ) { } ))



( ( { [ ) } { ] ) X



The no. of opening brackets = no. of closing brackets

↳ Infix, postfix, prefix evaluation

Stack ↑

↳ Conversion → Infix → Postfix | Prefix

Infix → a + b →



(( a \* b ) + ( c \* d )) / ( e + f )

prefix →  $\frac{a + b}{c * d}$

postfix → ab +

Syntax :-

int, Integer

Stack < Datatype > stack-name = new Stack<>();

→ 5 ~~wins~~

Ex:-

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

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

## ↳ Operations

a) Add element / Push

st.push(element)

LIFO      st.peek() → 70

b) size of stack

st.size()  
4

b) Removing element / Pop

st.pop()

c) Top most element / Peek

st.peek()



st.push(40)  
st.push(30)  
st.push(50)  
st.push(70)  
  
st.pop()  
st.pop()  
st.pop()  
→ st.pop()  
→ st.pop() → ~~stack underflow~~

→ ~~Stack overflow~~

→ adding an element to a already full stack

```
public static void main(String[] args) {  
    /* Enter your code here. Read input from STDIN  
     * Your class name should be Main  
     */  
    Stack<Integer> st = new Stack<>();  
  
    st.push(12);  
    st.push(23);  
    st.push(34);  
    st.push(45);  
  
    System.out.println(st.size());  
    System.out.println(st);  
    st.pop();  
    System.out.println(st);  
  
    while(st.size()>0){  
        System.out.println(st.pop());  
    }  
  
    st.pop();  
}
```

# Stack Syntax Learning Day 34

Problem

Submissions

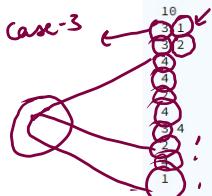
Leaderboard

Discussions

- Declare an Empty stack s .
- Take Single Integer T as input.
- For next T Lines format  
1. case 1. Print the size of the stack in a separate line.  
2. case 2. Remove an element from the stack. If the stack is empty then print in a separate line.  
3. case 3. Add Integer to the stack s.  
4. case 4. Print an element at the top of the stack. If stack is empty print in a seperate line.

$T = 10$       peak

Sample Input 0



Case → 3

Sample Output 0



1

s

- 1 → print size of st .
- 2 → Pop an element | → -1
- 3 → Push an element →
- 4 → Peak | → -1

```
public class Solution {
```

```
    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STD
        Scanner scn = new Scanner(System.in);
        int t = scn.nextInt();
        Stack<Integer> st = new Stack<>();
        for(int i=0;i<t;i++){
            int c = scn.nextInt();
            if(c == 1){
                System.out.println(st.size());
            }else if (c == 2){
                if(st.size() == 0) System.out.println(-1);
                else st.pop();
            }else if(c == 3){
                int x = scn.nextInt();
                st.push(x);
            }else{
                if(st.size() == 0) System.out.println(-1);
                else System.out.println(st.peek());
            }
        }
    }
```

→ 5 min

~~Monday~~ →  
~~Tuesday~~ →  
~~Wednesday~~ →

→ 9 - 11 pm

Saturday → 2<sup>nd</sup> sept

Knight tour

ArrayList, Stack, Object

int, char, Object, wrapper class

Integer, character

# Reverse string using stack (Day 34)

Problem

Submissions

Leaderboard

Discussions

Given a String str. We have to reverse the string str with help of only stack s.

**Input Format**

Single String str.

**Constraints**

$1 \leq \text{str.length()} \leq 10^5$

**Output Format**

Print the reverse of the String str.

**Sample Input 0**

```
abcdee
```

**Sample Output 0**

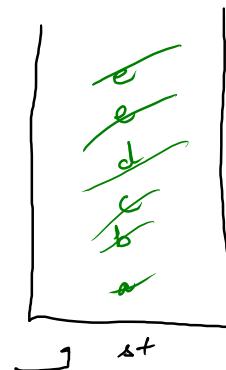
```
eedcba
```

$\text{str} = "abc\ dee"$

reversing

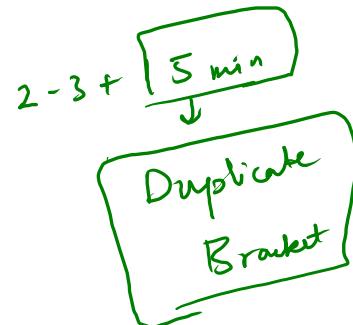
$\text{str} = \underline{\underline{\text{eedcba}}}$

eedcba



stack < character > st = new stack

```
public class Solution {  
  
    public static void main(String[] args) {  
        /* Enter your code here. Read input from STDIN. Pi  
        Scanner scn = new Scanner(System.in);  
        String str = scn.next();  
        Stack<Character> st = new Stack<>();  
  
        for(int i=0;i<str.length();i++){  
            st.push(str.charAt(i));  
        }  
  
        StringBuilder sb = new StringBuilder();  
        while(st.size()!=0){  
            sb.append(st.pop());  
        }  
  
        System.out.println(sb.toString());  
    }  
}
```



## Duplicate Brackets (Day 34)

### Problem

## Submissions

### Leaderboard

### Discussions

## Discussions

- c me of*

  1. You are given a string exp representing an expression.
  2. Assume that the expression is balanced i.e. the opening and closing brackets match with each other.
  3. But, some of the pair of brackets may be extra/needless.
  4. You are required to print true if you detect extra brackets and false otherwise.

e.g.  $((a + b) + (c + d)) \rightarrow \text{false}$   $(a + b) + ((c + d)) \rightarrow \text{true}$

$$\exp \rightarrow " \boxed{a+b} + \boxed{(c+d)} \rightarrow \underline{\text{time}}$$

$$\left( \begin{matrix} (a+b)^+ & (c+d)^+ \\ \uparrow \uparrow \uparrow \uparrow \uparrow \uparrow & \uparrow \uparrow \uparrow \end{matrix} \right)$$

$$\text{exp} \rightarrow ((\text{at}_b)(+)(\text{lt}_d)) \rightarrow \underline{\text{false}}$$

$\rightarrow$  symbol, alphabet, opening bracket  $\rightarrow$  push into stack

→ closing bracket

decision

st. peek() = = ( ) ~~return~~

→ { So to elements int'l you  
are <sup>not</sup> encounter with  
opening character.

of  $\mu$  is  
referred  
here  
this forecast  
is extra

st.  
↳ characters

String = "0"

→ Ultra fine

### Sample Input 0

$$(a + b) + ((c + d))$$

## Sample Output 0

true

Mou / Tuan / Wsd  
↓  
9 - 11 PM

```
public class Solution {  
    public static boolean duplicateBrackets(String str){  
        Stack<Character> st = new Stack<>(); →  
        for(int i=0;i<str.length();i++){  
            char ch = str.charAt(i);  
            if(ch != ')'){  
                st.push(ch);  
            }else{  
                if(st.peek() == '('){  
                    return true; →  
                }else{  
                    while(st.peek() != '('){  
                        st.pop();  
                    }  
                    st.pop();  
                }  
            }  
        }  
  
        return false;  
    }  
  
    public static void main(String[] args) {  
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */  
        Scanner scn = new Scanner(System.in);  
        String str = scn.nextLine();  
        System.out.println(duplicateBrackets(str));  
    }  
}
```

→ 5 min  
i = 0 X 4 / 3  
X  
\$6  
a = x axb xfc ( )  
  