

[◀ Array](#) [Matrix](#) [Strings](#) [Hashing](#)[Linked List](#) [Stack](#) [Queue](#) [Binary Tree](#) [Binary Search Tree](#) [Heap](#) [Graph](#) [Searching](#) [Sorting](#) [Divide & Conquer](#) [Mathematical](#) [Geometric](#) [Bitwise](#) ➤

Array Data Structure

Introduction to Arrays

Searching ▾

Sorting ▾

Rotation ▾

Arrangement and rearrangement ▾

Order Statistics ▾

Range queries ▾

Optimization ▾

Some Other Interesting Problems ▾

Subarray/Substring vs  
Subsequence and Programs to  
Generate them

Product of Array except itself

## Implement two stacks in an array

Difficulty Level : Easy • Last Updated : 04 Jul, 2022



Create a data structure `twoStacks` that represents two stacks. Implementation of `twoStacks` should use only one array, i.e., both stacks should use the same array for storing elements.

*Following functions must be supported by `twoStacks`.*

- `push1(int x)` -> pushes `x` to first stack
- `push2(int x)` -> pushes `x` to second stack
- `pop1()` -> pops an element from first stack and return the popped element
- `pop2()` -> pops an element from second stack and return the popped element

*Implementation of `twoStack` should be space efficient.*

Recommended Practice

**Implement two stacks in an array****Try It!**

### Complete Interview Preparation - Self Paced

By Sandeep Jain

Beginner to Advance Level

Find 360 solution to all of your interview woes. Learn 4 years' worth of programming knowledge in just 6 months and ace coding interviews at top tech companies.

**Explore Now**

#### Method 1 (Divide the space in two halves):

A simple way to implement two stacks is to divide the array in two halves and assign the half space to two stacks, i.e., use `arr[0]` to `arr[n/2]` for stack1, and `arr[(n/2) + 1]` to `arr[n-1]` for stack2 where `arr[]` is the array to be used to implement two stacks and size of array be `n`. The problem with this method is inefficient use of array space. A stack push operation may result in stack overflow even if there is space available in `arr[]`. For example, say the array size is 6 and we push 3 elements to stack1 and do not push anything to second stack2. When we push 4th element to stack1, there will be overflow even if we have space for 3 more elements in array.

[C++](#)[Java](#)[Python3](#)[C#](#)[Javascript](#)

```
# Python Script to Implement two stacks in a list
import math

class twoStacks:

    def __init__(self, n):          # constructor
        self.size = n
        self.arr = [None] * n
        self.top1 = math.floor(n/2) + 1
        self.top2 = math.floor(n/2)

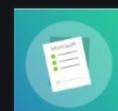
    # Method to push an element x to stack1
    def push1(self, x):

        # There is at least one empty space for new element
        if self.top1 > 0:
            self.top1 = self.top1 - 1
            self.arr[self.top1] = x
        else:
            print("Stack Overflow by element : ", x)

    # Method to push an element x to stack2
    def push2(self, x):

        # There is at least one empty space for new element
```

#### WHAT'S NEW



**Microsoft SDE Preparation Test Series**

[View Details](#)

**Data Structures & Algorithms- Self Paced Course**

[View Details](#)

**Complete Interview Preparation- Self Paced Course**

[View Details](#)

```

        if self.top2 < self.size - 1:
            self.top2 = self.top2 + 1
            self.arr[self.top2] = x
        else :
            print("Stack Overflow by element : ", x)

    # Method to pop an element from first stack
    def pop1(self):
        if self.top1 <= self.size/2:
            x = self.arr[self.top1]
            self.top1 = self.top1 +1
            return x
        else:
            print("Stack Underflow ")
            exit(1)

    # Method to pop an element from second stack
    def pop2(self):
        if self.top2 >= math.floor(self.size/2) + 1:
            x = self.arr[self.top2]
            self.top2 = self.top2 - 1
            return x
        else:
            print("Stack Underflow ")
            exit(1)

    # Driver program to test twoStacks class
ts = twoStacks(5)
ts.push1(5)
ts.push2(10)
ts.push2(15)
ts.push1(11)
ts.push2(7)

print("Popped element from stack1 is : " + str(ts.pop1()))
ts.push2(40)
print("Popped element from stack2 is : " + str(ts.pop2()))

# This code is contributed by Gautam goel

```

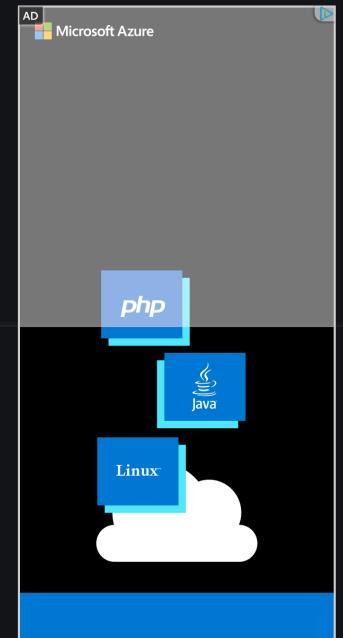
### Output

```

Stack Overflow By element :7
Popped element from stack1 is  : 11
Stack Overflow By element :40

Popped element from stack2 is : 15

```



### Complexity Analysis:

- **Time Complexity:**
    - **Push operation :** O(1)
    - **Pop operation :** O(1)
  - **Auxiliary Space:** O(N).
- Use of array to implement stack so. It is not the space-optimized method as explained above.

### Method 2 (A space efficient implementation) :

This method efficiently utilizes the available space. It doesn't cause an overflow if there is space available in arr[]. The idea is to start two stacks from two extreme corners of arr[]. stack1 starts from the leftmost element, the first element in stack1 is pushed at index 0. The stack2 starts from the rightmost corner, the first element in stack2 is pushed at index (n-1). Both stacks grow (or shrink) in opposite direction. To check for overflow, all we need to check is for space between top elements of both stacks. This check is highlighted in the below code.

C++ Java Python C# PHP Javascript

```

# Python Script to Implement two stacks in a list
class twoStacks:

    def __init__(self, n):      # constructor
        self.size = n
        self.arr = [None] * n
        self.top1 = -1
        self.top2 = self.size

    # Method to push an element x to stack1
    def push1(self, x):

        # There is at least one empty space for new element
        if self.top1 <= self.top2 - 1:

```

```

        if self.top1 < self.top2 - 1 :
            self.top1 = self.top1 + 1
            self.arr[self.top1] = x

        else:
            print("Stack Overflow ")
            exit(1)

    # Method to push an element x to stack2
    def push2(self, x):

        # There is at least one empty space for new element
        if self.top1 < self.top2 - 1:
            self.top2 = self.top2 - 1
            self.arr[self.top2] = x

        else:
            print("Stack Overflow ")
            exit(1)

    # Method to pop an element from first stack
    def pop1(self):
        if self.top1 >= 0:
            x = self.arr[self.top1]
            self.top1 = self.top1 -1
            return x
        else:
            print("Stack Underflow ")
            exit(1)

    # Method to pop an element from second stack
    def pop2(self):
        if self.top2 < self.size:
            x = self.arr[self.top2]
            self.top2 = self.top2 + 1
            return x
        else:
            print("Stack Underflow ")
            exit()

    # Driver program to test twoStacks class
    ts = twoStacks(5)
    ts.push1(5)
    ts.push2(10)
    ts.push2(15)
    ts.push1(11)
    ts.push2(7)

    print("Popped element from stack1 is " + str(ts.pop1()))
    ts.push2(40)
    print("Popped element from stack2 is " + str(ts.pop2()))

    # This code is contributed by Sunny Karira

```

#### Output:

```

Popped element from stack1 is 11
Popped element from stack2 is 40

```

#### Complexity Analysis:

- **Time Complexity:**

- **Push operation :** O(1)
- **Pop operation :** O(1)

- **Auxiliary Space :** O(N).

Use of array to implement stack so it is a space-optimized method.

 **Method 2 (A space efficient implementation)** 

This video covers the implementation of two stacks in an array. It discusses how to start two stacks from two extreme corners of the array and how they grow in opposite directions. The video also highlights the space-optimized nature of this implementation.

**Key Points:**

- Efficient utilization of available space.
- No overflow occurs if there is space available in the array.
- The idea is to start two stacks from two extreme corners of the array.
- stack1 starts from the leftmost element, the first element in stack1 is pushed at index 0.
- stack2 starts from the rightmost corner, the first element in stack2 is pushed at index (n-1).
- Both stacks grow (or shrink) in opposite direction.
- To check for overflow, all we need to check is for space between top elements of both stacks.

Watch on  YouTube

 Like 186

< Previous

Find whether an array is subset  
of another array

Next >

Find relative complement of two  
sorted arrays



#### RECOMMENDED ARTICLES

Page : 1 2 3

01 How to efficiently implement k stacks in a single array?  
22, Aug 14

05 Reversing a Stack using two empty Stacks  
09, Jun 21

02 Implement Dynamic Multi Stack (K stacks) using only one Data Structure  
09, Dec 21

06 Maximum count of integers to be chosen from given two stacks having sum at most K  
24, Jan 22

03 Infix to Prefix conversion using two stacks  
17, Jun 18

07 Merging and Sorting Two Unsorted Stacks  
20, May 18

04 Add two numbers represented by Stacks  
20, Aug 20

08 Iterative Postorder Traversal | Set 1 (Using Two Stacks)  
23, Feb 13

• • •

#### Article Contributed By :



#### Vote for difficulty

Current difficulty : [Easy](#)

[Easy](#) [Normal](#) [Medium](#) [Hard](#) [Expert](#)

avonitrachhadiya2155, krishna\_97, umadevi9616, rkbhola5, Rajput-Ji, simmytarika5, gautamgoel962, hardikkoriintern

**Article Tags :** 24\*7 Innovation Labs, Accolite, Microsoft, Samsung, Snapdeal, Arrays, Stack

**Practice Tags :** 24\*7 Innovation Labs, Accolite, Microsoft, Samsung, Snapdeal, Arrays, Stack

[Improve Article](#)

[Report Issue](#)

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

[Load Comments](#)

AD



All your transactions.  
All in one place.



◎ A-143, 9th Floor, Sovereign Corporate Tower,  
Sector-136, Noida, Uttar Pradesh - 201305

✉ feedback@geeksforgeeks.org



#### Company

About Us

Careers

In Media

Contact Us

Privacy Policy

Copyright Policy

#### Learn

Algorithms

Data Structures

SDE Cheat Sheet

Machine learning

CS Subjects

Video Tutorials

Courses

#### News

Top News

Technology

Work & Career

Business

Finance

Lifestyle

Knowledge

#### Languages

Python

Java

CPP

Golang

C#

SQL

Kotlin

#### Web Development

Web Tutorials

Django Tutorial

HTML

JavaScript

Bootstrap

ReactJS

NodeJS

#### Contribute

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships

Video Internship

Microsoft Build and deploy Node.js, Java, and Python apps with Azure.

[LEARN MORE](#)

HIDE AD • AD VIA BUYSSELLADS