

GeeksforGeeks

A computer science portal for geeks

[Android App](#) [GeeksQuiz](#)

[Login/Register](#)

- [Home](#)
- [Algorithms](#)
- [DS](#)
- [GATE](#)
- [Interview Corner](#)
- [Q&A](#)
- [C](#)
- [C++](#)
- [Java](#)
- [Books](#)
- [Contribute](#)
- [Ask a Q](#)
- [About](#)

[Array](#)

[Bit Magic](#)

[C/C++](#)

[Articles](#)

[GFacts](#)

[Linked List](#)

[MCQ](#)

[Misc](#)

[Output](#)

[String](#)

[Tree](#)

[Graph](#)

Implement two stacks in an array

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.

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 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.

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.

```
#include<iostream>
#include<stdlib.h>

using namespace std;

class twoStacks
{
    int *arr;
    int size;
    int top1, top2;
public:
    twoStacks(int n) // constructor
    {
        size = n;
        arr = new int[n];
        top1 = -1;
        top2 = size;
    }

    // Method to push an element x to stack1
    void push1(int x)
    {
        // There is at least one empty space for new element
        if (top1 < top2 - 1)
        {
            top1++;
            arr[top1] = x;
        }
        else
        {
            cout << "Stack Overflow";
            exit(1);
        }
    }
}
```

```
}

// Method to push an element x to stack2
void push2(int x)
{
    // There is at least one empty space for new element
    if (top1 < top2 - 1)
    {
        top2--;
        arr[top2] = x;
    }
    else
    {
        cout << "Stack Overflow";
        exit(1);
    }
}

// Method to pop an element from first stack
int pop1()
{
    if (top1 >= 0 )
    {
        int x = arr[top1];
        top1--;
        return x;
    }
    else
    {
        cout << "Stack UnderFlow";
        exit(1);
    }
}

// Method to pop an element from second stack
int pop2()
{
    if (top2 < size)
    {
        int x = arr[top2];
        top2++;
        return x;
    }
    else
    {
        cout << "Stack UnderFlow";
        exit(1);
    }
}

};

/* Driver program to test twStacks class */
```

```
int main()
{
    twoStacks ts(5);
    ts.push1(5);
    ts.push2(10);
    ts.push2(15);
    ts.push1(11);
    ts.push2(7);
    cout << "Popped element from stack1 is " << ts.pop1();
    ts.push2(40);
    cout << "\nPopped element from stack2 is " << ts.pop2();
    return 0;
}
```

Output:

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

Time complexity of all 4 operations of *twoStack* is $O(1)$.
We will extend to 3 stacks in an array in a separate post.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Topics:

- [Count number of islands where every island is row-wise and column-wise separated](#)
- [Rearrange an array such that 'arr\[j\]' becomes 'i' if 'arr\[i\]' is 'j'](#)
- [Find position of an element in a sorted array of infinite numbers](#)
- [Find a common element in all rows of a given row-wise sorted matrix](#)
- [Can QuickSort be implemented in \$O\(n \log n\)\$ worst case time complexity?](#)
- [Fill two instances of all numbers from 1 to n in a specific way](#)
- [Weighted Job Scheduling](#)
- [Factorial of a large number](#)

Tags: [stack](#)

Like < 12

Tweet < 0

 < 3

Writing code in comment? Please use [ideone.com](#) and share the link here.

46 Comments

GeeksforGeeks

 Login ▾

 Recommend 2  Share

Sort by Newest ▾



Join the discussion...



Kerem Sahin · 3 months ago



My implementation in C

<http://ideone.com/e95RIA>

^ | v • Reply • Share ›



Chanchal • 3 months ago

if my stack1 pushes all its elements into even positions of the array and stack2 pushes them into odd functions... the program would work and be space efficient as well. Then we dont need to define the size of the array. We can define the array dynamically.

^ | v • Reply • Share ›



instance → Chanchal • 3 months ago

No Chanchal, lets say array size = 10, in case you push first 6 elements in stack1 (nothing in stack2), it will overflow for 6th element since each stack can accomodate 5 elements. Your solution is equivalent to Scenario 1 mentioned 1 which is inefficient.

1 ^ | v • Reply • Share ›



Chain Singh • 4 months ago

```
package com.bt.h2;
```

```
public class Stack {
```

```
    int top1 = -1;
```

```
    int top2 = 10;
```

```
    int arr[] = new int[10];
```

```
    void push(int stackno , int i){
```

```
        if(top1 == top2 -1)
```

```
            throw new IllegalStateException("stack is full");
```

```
        if(stackno == 1)
```

```
            arr[++top1] = i;
```

```
        else
```

[see more](#)

1 ^ | v • Reply • Share ›



Guest • 6 months ago

```
#include "stdafx.h"
```

```
#include <exception>
```

```
using namespace std;

struct DualStack
{
    unsigned int _capacity;

    int topStack1;

    int topStack2;

    int* stackArray;

    DualStack(unsigned int capacity)
```

```
{
```

[see more](#)

^ | v • Reply • Share ›



Mohit Chaudhary • 6 months ago

```
#include<stdio.h>
#include<stdlib.h>
struct twostacks{
    int top1;
    int top2;
    unsigned int capacity;
    int* array;
};

struct twostacks* create(unsigned int capacity)
{
    struct twostacks* twostacks=(struct twostacks*)malloc(sizeof(struct twostacks));
    twostacks->top1=-1;
    twostacks->capacity=capacity;
    twostacks->top2=twostacks->capacity;
    twostacks->array=(int*)malloc(twostacks->capacity * sizeof(int));
    return twostacks ;
}
```

[see more](#)

^ | v • Reply • Share ›



devakar verma → Mohit Chaudhary • a month ago

In function isFull , the difference between two tops should be equal to 1.

top2-top1==1

for eg capacity=2, after 1 push in both stack, results top1=0 and top2=1. Here array will be full.

^ | v • Reply • Share ›



pango89 • 6 months ago

Please Look at this Post for Implementing K Stacks in One Array.

<http://www.geeksforgeeks.org/e...>

^ | v • Reply • Share ›



spiderweb • 6 months ago

please explain implementing three stacks

^ | v • Reply • Share ›



pango89 → spiderweb • 6 months ago

Please Look at this Post

<http://www.geeksforgeeks.org/e...>

2 ^ | v • Reply • Share ›



spiderweb → pango89 • 6 months ago

thnx :)

^ | v • Reply • Share ›



Kim Jong-il → spiderweb • 6 months ago

For 3-stack or more than that i.e. n-stack, we can use only first method.

second method was only applicable if we have to implement two stack.

^ | v • Reply • Share ›



Lex • 7 months ago

How do I display the values in each stack from your code?

^ | v • Reply • Share ›



Rashid khan • 7 months ago

@GeeksForGeeks how to implement k stacks in a given array?

1 ^ | v • Reply • Share ›



Vaibhav Jain → Rashid khan • 7 months ago

Go for method 1 given above. divide array in k equal parts and fill it accordingly.

^ | v • Reply • Share ›



Paparao Veeragandham • 8 months ago



Hi Geeksforgeeks,
can please explain implementing 3 stacks into array effectively by space

1 ^ | v • Reply • Share ›



Gaurav • 9 months ago

Java code for this problem

<http://ideone.com/ctxm9u>

^ | v • Reply • Share ›



Sai Krishna • 9 months ago

For implementing k stacks in an array----->

Let us consider an array of size n, and for i^{th} stack top element is $n/k(i-1)$,
i.e for stack-1 $\text{top}_1=0$, for stack-2 $\text{top}_2=n/k$, lly for others.

and while pushing elements in i^{th} stack u have to check whether its top element is equals to the base reference of $(i+1)$ stack i.e if its not then u can push directly otherwise it requires shifting of the elements of $(i+1)$ stack to right side provided that if stack $(i+1)$ is not full lly check the validations for others too-----

^ | v • Reply • Share ›



Ayush Jain → Sai Krishna • 8 months ago

Are you saying that we should divide an array into k parts?

Suppose $k=3$ and $n=9$.

Stack1: $i=0$ to $i=2$

Stack2: $i=3$ to $i=5$

Stack3: $i=6$ to $i=8$

Is this what you are saying?

^ | v • Reply • Share ›



np → Ayush Jain • 8 months ago

I think yes and addition to that suppose my 1st stack is full then we will check second is full or not if it is not full then move 2nd stack forward by 1 element to accommodate an element from stack 1. If in case stack 2 is full check for space in stack 3 and so on.....

^ | v • Reply • Share ›



tushar → Sai Krishna • 9 months ago

will you please elaborate

^ | v • Reply • Share ›



coder • 10 months ago

@GeeksForGeeks how to implement k stacks in a given array?

^ | v • Reply • Share ›

**Kim Jong-il** → coder · 6 months ago

By using First method.

^ | v · Reply · Share ›

**Arulmozhi** · a year ago

how about storing elements alternately. This will implement 2 stacks in a array.

^ | v · Reply · Share ›

**仮面 の男** → Arulmozhi · a year ago

With the additional restriction that neither stack should overflow unless the total number of elements is n, your idea wouldn't work, but if not, then it is okay.

^ | v · Reply · Share ›

**Arifa Khan** · 2 years ago

that's not good.

^ | v · Reply · Share ›

**abhishek08aug** · 2 years ago

```
#include<stdio.h>

struct two_stacks {
    int first_stack_top;
    int second_stack_top;
    int size;
    int * array;
};

void push(struct two_stacks * ts, int insert_value, int stack_num) {
    if(stack_num==1) {
        if(ts->first_stack_top+1==ts->second_stack_top) {
            printf("ERROR: two stack array full! can't have any more elements!\n");
            return;
        } else {
            ts->first_stack_top++;
            ts->array[ts->first_stack_top]=insert_value;
        }
    }
}
```

[see more](#)

1 ^ | v · Reply · Share ›

**Ganesh** · 2 years ago

You can find java code here:

`[sourcecode language="JAVA"]`

```

/**
 * 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.
 *
 * @author GAPIITD
 *
 */
public class TwoStack {
    private int stack[];
    private int top1, top2;

    TwoStack() {
        this(10);
    }

```

[see more](#)

1 ^ | v • Reply • Share ›

**abhay** • 2 years ago

using an array to store the even and odd numbers...

```

#include<stdio.h>
#include<stdlib.h>

struct node
{
    int data;
    struct node* next;
};

void insertbeg(int value, struct node **head)
{
    struct node *temp=malloc(sizeof(struct node));
    temp->data=value;

```

[see more](#)

^ | v • Reply • Share ›

**Geek** • 2 years ago

Even Better Memory optimized solution..

1) push everything from start
2) Create link lists inside array..
say input order:
which stack it should go
stack 1 : 10
stack 2 : 12
stack 2 : 16
stack 1 : 1
stack 2 : 23
answer is:



Always Maintain 4 pointers
stack1Head, stack1Tail
stack1Head stack2Tail

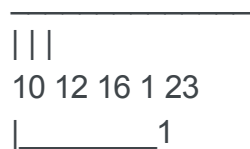
obvs you can get idea about popping.

```
[sourcecode language=""]
/* Paste your code here (You may delete these lines if not writing code) */
^ | v • Reply • Share ›
```



Geek  Geek • 2 years ago

/*



*/

^ | v • Reply • Share ›



Geek Geek • 2 years ago

Not able to fix this indentation...

lists will be

12->16->23

16->1

1 ^ | v • Reply • Share ›

**Thirunavukkarasu** · 3 years ago

```
#include<stdio.h>

int arr[10];
int top1=-1,top2=sizeof(arr)/sizeof(arr[0]);
void push1(int data)
{
    if(top1+1==top2)
    { printf("Stack is full "); return;}
    else
    {
        top1++;
        arr[top1]=data;
    }
}

void push2(int data)
{
    if(top1==top2-1)
    {
```

[see more](#)

1 ^ | v · Reply · Share ›

**Seema** · 3 years ago

Great work! Keep it on!

^ | v · Reply · Share ›

**Krupa** · 3 years ago

Making 2-stacks in the method(2) presented above is nothing but how OS can manage the stack and heap both growing in reverse direction. Its simple solution.

1 ^ | v · Reply · Share ›

**test** · 3 years ago

test

^ | v · Reply · Share ›

**Inderpreet Singh** · 3 years ago

Best Stuff. It's very easy to understand. I tried to understand this concept many times before this post was written but i could not. After reading this post, i understood it . Thanks Man and Thanks GeeksforGeeks.

^ | v · Reply · Share ›

**suhas meena** · 3 years ago



In Second method while we do pop operation. Their is a case that while we pop stack 2 items we may pop stack 1 first item because we are not differentiating between items of two stack. Correct me if i am wrong.

```
/* Paste your code here (You may delete these lines if not writing code) */
```

^ | v • Reply • Share ›



kartik → suhas meena • 3 years ago

That is not possible because there are two different top variables and different conditions to check underflow.

^ | v • Reply • Share ›



kunal • 3 years ago

you people provide the best stuff to crack interview

keep it on :)

1 ^ | v • Reply • Share ›



laxmi • 3 years ago

like your explanation

^ | v • Reply • Share ›



Other Neo • 3 years ago

The code look amateur "C" code written at the very best....

1. push1 and push2 do not return the error/exception to the caller
2. Big Blunder in terms of EXIT(0)
3. Duplicate code with respect to push1 & push2 and pop1 and pop2. Ideally push and pop should take the parameter about the stack on which to operate

^ | v • Reply • Share ›



kartik → Other Neo • 3 years ago

@Other Neo: Thanks for your inputs. The coding is done this way to keep code small and simple. Could you provide more details or reference about the problem with exit(0).

Also, all code in push1() and push2() not duplicatae, it's just the overflow handling part. The push1() operations does top1++ and push2() does top1--. Similarly pop1() and pop2() are not duplicate.

^ | v • Reply • Share ›



Other Neo → kartik • 3 years ago

@Kartik "exit(0)" are old rudimentary ways for error handling done by really old code. exit(0) in any code just shows the arrogance of the code / programmer. It breaks all the call chain (behaving more like goto) and closes the program

breaks all the call chain (behaving more like goto) and closes the program without any possibility of recovery.

In your code the instead of EXIT(0) you could have simply written RETURN FALSE and thus provide the opportunity to the caller of the function (main) to deal with the error.

PUSH functions are exact duplicates of each other if we ignore the comparison and increment/decrement; and the same goes with POP. For illustration if the implementation was to change from a static array to std::vector, you would need to change code twice for POP and twice for PUSH.

If you are still not convinced let me know and I will rewrite the whole code.

^ | v • Reply • Share ›



lalor → Other Neo • 3 years ago

I hope you give the whole code.

```
/* Paste your code here (You may delete these lines if not writing c
```

^ | v • Reply • Share ›



Ashakiran Bhatler → lalor • 4 months ago

Dividing the array into three stacks :)

// by ashakiran bhatler

// compile: g++ -std=c++11 test.cpp

// run : ./a.out

// sample output as below

// adding: 1 2 3 4 5 6 7 8 9

// array contents: 9 8 7 6 5 4 3 2 1

// popping now...

// array contents: 8 7 6 5 4 3 2 1

#include <iostream>

#include <cstdlib>

[see more](#)

1 ^ | v • Reply • Share ›



Subscribe



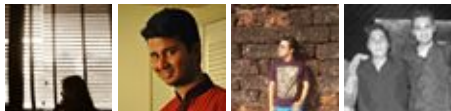
Add Disqus to your site



Privacy

**GeeksforGeeks**

Like

92,220 people like GeeksforGeeks.

Facebook social plugins

- [Interview Experiences](#)
- [Advanced Data Structures](#)
- [Dynamic Programming](#)
- [Greedy Algorithms](#)
- [Backtracking](#)
- [Pattern Searching](#)
- [Divide & Conquer](#)
- [Mathematical Algorithms](#)
- [Recursion](#)
- [Geometric Algorithms](#)

• Popular Posts

- [All permutations of a given string](#)
- [Memory Layout of C Programs](#)
- [Understanding “extern” keyword in C](#)
- [Median of two sorted arrays](#)
- [Tree traversal without recursion and without stack!](#)
- [Structure Member Alignment, Padding and Data Packing](#)
- [Intersection point of two Linked Lists](#)
- [Lowest Common Ancestor in a BST](#)
- [Check if a binary tree is BST or not](#)
- [Sorted Linked List to Balanced BST](#)

Follow @GeeksforGeeks



[Subscribe](#)

• Recent Comments

- [guest](#)

how is this problem different from LISS question

[Vertex Cover Problem | Set 2 \(Dynamic Programming Solution for Tree\)](#) · [6 minutes ago](#)

- DD

void delLesserNodes(struct node *head) { struct...

[Delete nodes which have a greater value on right side](#) · [13 minutes ago](#)

- [Jason Bourne](#)

got it, Thanks!

[Maximum size square sub-matrix with all 1s](#) · [28 minutes ago](#)

- [Jason Bourne](#)

What shall be the max sum path if the tree has...

[Find the maximum path sum between two leaves of a binary tree](#) · [32 minutes ago](#)

- Anotherone

Hm. Actually, we can do better. Approximately,...

[How to print maximum number of ‘A’ using given four keys](#) · [1 hour ago](#)

- [Aditya Goel](#)

I don't understand why Step-2 is divided into...

[Boundary Traversal of binary tree](#) · [1 hour ago](#)

•

@geeksforgeeks, [Some rights reserved](#) [Contact Us!](#)

Powered by [WordPress](#) & [MooTools](#), customized by geeksforgeeks team