

GeeksforGeeks

A computer science portal for geeks

Introduction:

As we know, there are now so many public domain learning materials on the internet, which are also known as Open Education Resources (OERs). It is favourable that there are a large number of online resources for us to learn. However, the quality of online resources varies a lot. This time I want to share with you a really useful website for computer science students, especially in data structures. Because of this website, **I got an A+ in the Data Structure Course (COMP2021)**. Many of my friends also highly recommend this website and I believe it will help you if you are having trouble with data structures.

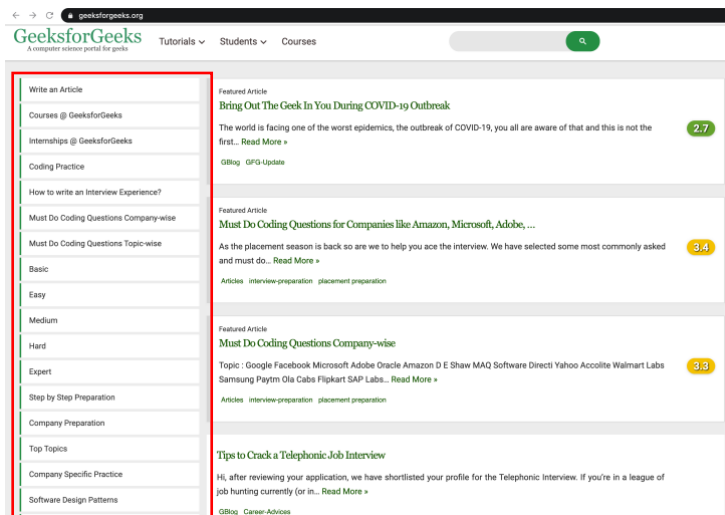
Brief Reasons of Why I Recommend GeeksforGeeks:

The reasons mainly contains three parts:

1. It combines all kinds of resources about computer science and saves students' time searching them on the internet.
2. It is especially useful to learn data structures and algorithms, because it provides a huge number of high frequency questions and exercises for each data structures with clear explanations.
3. Its solutions are easy to understand and include multiple text solutions, online practice, video explanation, and codes for different languages.

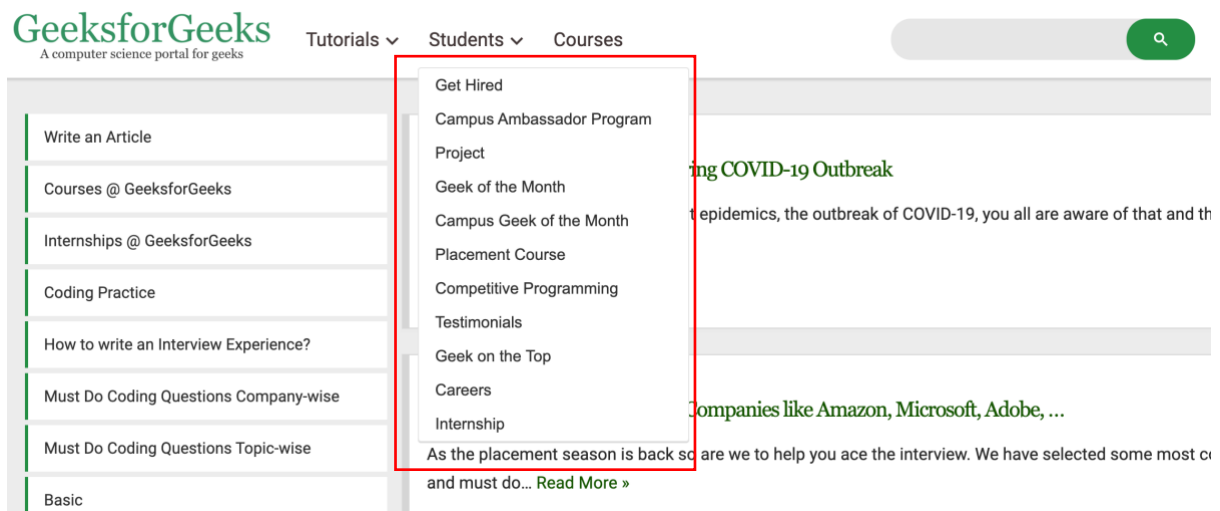
From my research, online resources mainly contain three categories, which are online videos/courses, online tutorials, and online programming websites. For example, you can get a lot of useful videos from [YouTube](#), plenty of tutorials of programming languages in [W3Schools](#) and over one thousand exercises in [LeetCode](#). However, the websites that can combine all three of them are very rarely----[GeeksforGeeks](#) is one of them.

This passage will not only tell you why to use GeeksforGeeks but also tell you how to use it! Let's first have a look at the main page of the website. You can find probably all the computer science topics on the right-hand side of the page, which also includes many internship experiences and learning experiences. You can also write your article here if you have some unique solutions to the questions!



Expert	Leaderboard !!
Step by Step Preparation	Topic-wise Practice
Company Preparation	Subjective Problems
Top Topics	Difficulty Level - School
Company Specific Practice	Difficulty Level - Basic
Software Design Patterns	Difficulty Level - Easy
Placements Preparation Course	Difficulty Level - Medium
Interview Corner	Difficulty Level - Hard
Recent Interview Experiences	Explore More...
GQ Home Page	C
Quiz Corner	C++
LMNs	Java
What's New ?	Python
Leaderboard !!	SQL
Topic-wise Practice	PHP
Subjective Problems	JavaScript
Difficulty Level - School	School Programming
Difficulty Level - Basic	Operating Systems
Difficulty Level - Easy	DBMS

For different students, it also provides different parts to satisfy different people's needs.

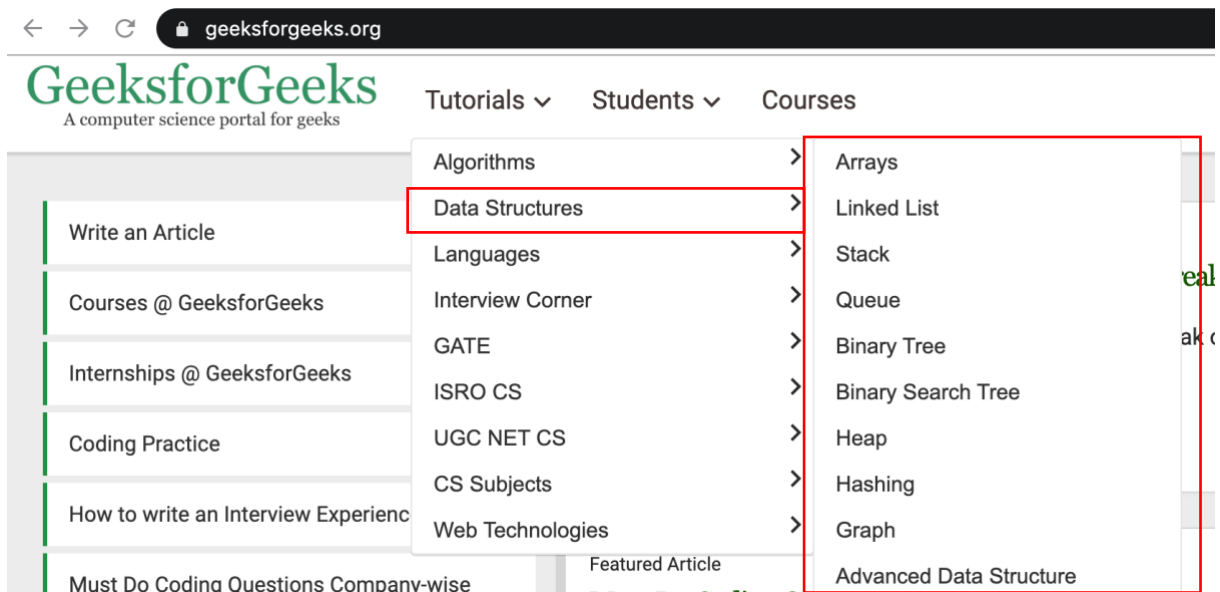


About the courses, I do not recommend them because there are so many high-quality online courses on other websites.

All in all, GeeksforGeeks can satisfy nearly all of computer science's basic requirements. If you have this website, it can save you much time searching for some topic on the internet from a huge amount of good or bad resources. That's why I highly recommend it. Next, I will introduce how to use it to practice your data structures, which is maybe one of the most important topics in the CS college study.

How to learn data structures in GeeksforGeeks?

First, let's see how to find the tutorials of data structures. You can also choose the specific data structure that you want to practice.

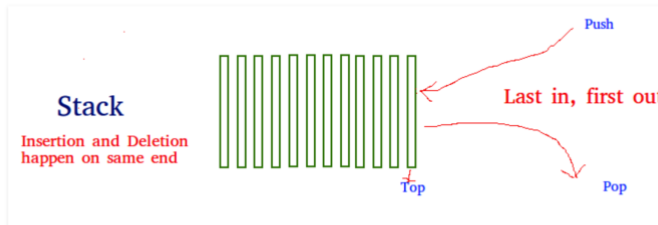


It will firstly introduce the features of the data structure and provide lots of questions about this data structure. For example, if we click “Stack”, we can find the following information.

Stack Data Structure

Recent articles on Stack

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO (Last In First Out) or FILO (First In Last Out).



There are many real-life examples of a stack. Consider an example of plates stacked over one another in the canteen. The plate which is at the top is the first one to be removed, i.e. the plate which has been placed at the bottommost position remains in the stack for the longest period of time. So, it can be simply seen to follow LIFO (Last In First Out)/FILO (First In Last Out) order.

Topics :

Introduction :

1. Introduction to Stack
2. Stack in C++ STL
3. Stack Class in Java
4. Stack in Python
5. Stack in C#

Design and Implementation :

1. Implement Queue using Stacks
2. [Design and Implement Special Stack Data Structure | Added Space Optimized Version](#)
3. [Implement two stacks in an array](#)
4. Implement Stack using Queues
5. Design a stack with operations on middle element
6. How to efficiently implement k stacks in a single array?
7. How to create mergable stack?
8. [Design a stack that supports getMin\(\) in O\(1\) time and O\(1\) extra space](#)
9. Implement a stack using single queue
10. How to implement stack using priority queue or heap?
11. Create a customized data structure which evaluates functions in O(1)
12. Implement Stack and Queue using Deque

Standard Problems based on Stack :

1. [Infix to Postfix Conversion using Stack](#)
2. Prefix to Infix Conversion
3. Prefix to Postfix Conversion
4. Postfix to Prefix Conversion
5. Postfix to Infix
6. Convert Infix To Prefix Notation
7. [The Stock Span Problem](#)
8. Check for balanced parentheses in an expression
9. Next Greater Element
10. Next Greater Frequency Element
11. Number of NGEs to the right
12. Maximum product of indexes of next greater on left and right
13. The Celebrity Problem

From my experience, the red topics above are the most important in this data structure and **it is very likely to be tested in all kinds of tests, interviews, and examinations**. If you can master all of them, it is a piece of cake to get a good grade in the data structure course. Now let's see how to learn every question in the above list.

How to learn a specific question?

Let's take "Implement two stacks in an array" as an example.

The screenshot shows the GeeksforGeeks website. The header includes the logo, navigation links (Tutorials, Students, Courses), and a search bar. A sidebar on the left lists several problems, with 'Implement two stacks in an array' highlighted in green. The main content area displays the title of the article, a brief description, and a list of functions to be implemented: `push1(int x)`, `push2(int x)`, `pop1()`, and `pop2()`.

It will first introduce the question and give you some examples. After understanding the scenarios, it is highly recommended to practice by ourselves first. You can either use your compiler or use the online version provided by the website. The link is highlighted in the following picture.

This screenshot shows a different part of the article. It includes a sidebar with various problem titles. The main content area contains the text 'Implementation of twoStack should be space efficient.' followed by a green recommendation: 'Recommended: Please solve it on "PRACTICE" first, before moving on to the solution.' This recommendation is enclosed in a red rectangular box. Below this, the article introduces 'Method 1 (Divide the space in two halves)' and explains how to divide the array into two halves for the two stacks.

After practicing by ourselves, it comes to the most significant part when using the GeeksforGeeks website. You can find **multiple solutions** to this question and find the simplest one. Usually, if it is an algorithm question, it will also provide **the time and space complexity of every solution**. That's so friendly for the beginners in data structures!

Find a pair (n,r) in an integer array such that value of nCr is maximum
Choose X elements from A[] and Y elements from B[] which satisfy the given condition
Minimum halls required for class scheduling
Check whether N is a Factorion or not
Index of the elements which are equal to the sum of all succeeding elements
C++ Program to print an Array using Recursion
Sum of values of all possible non-empty subsets of the given array
Count equal element pairs in the given array

Recommended: Please solve it on "PRACTICE" first, before moving on to the solution.

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.

It also provides the implementation of each solution in many different programming languages. Compare to many websites which only give solution but no implementation, it will save you a lot of time.

Bitwise XOR of elements having odd frequency

Sentinel Linear Search

How to implement our own Vector Class in C++?

Find the minimum value of X for an expression

Range maximum query using Sparse Table

Queries to find the count of integers in a range that contain the given pattern

Count the pairs in an array such that the

C++ Java Python C# PHP

```

#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;
    }

```

Some people may ask: what if I cannot understand the solutions? Don't worry. Usually, the website will provide a video solution to each question. Compared to the text version, the video version may be easier to understand especially in some sorting algorithms.

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.

EG

Implement TwoStacks in an array | GeeksforGeeks

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.

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



Video solution
under the merge
sort algorithm.

Conclusion:

In conclusion, GeeksforGeeks is the best **OER** I have ever seen in the CS area, especially for data structures and algorithms. It is also highly recognized by **Dr. Cao**, the lecturer of COMP2021. For me, I got a very ideal grade using this website and all the students who have used this website also considered that it is very useful. Therefore, I sincerely recommend this website to all the COMP students who want to improve their data structures skills no matter for the sake of examinations or the interviews.

ZHANG Caiqi