# Stack Data Structure

[**Recent Articles**](https://www.geeksforgeeks.org/category/stack/)  
**Topic :**

* [Introduction](https://www.geeksforgeeks.org/stack-data-structure/#intro)
* [Design and Implementation](https://www.geeksforgeeks.org/stack-data-structure/#design)
* [Standard Problems](https://www.geeksforgeeks.org/stack-data-structure/#standard)
* [Operations on Stack](https://www.geeksforgeeks.org/stack-data-structure/#operations)
* [Misc](https://www.geeksforgeeks.org/stack-data-structure/#misc)
* [Quick Links](https://www.geeksforgeeks.org/stack-data-structure/#quick)

**Introduction :**

1. [Introduction to Stack](https://www.geeksforgeeks.org/stack-data-structure-introduction-program/)
2. [Stack in C++ STL](https://www.geeksforgeeks.org/stack-in-cpp-stl/)
3. [Stack Class in Java](https://www.geeksforgeeks.org/stack-class-in-java/)

**Design and Implementation :**

1. [Implement Queue using Stacks](https://www.geeksforgeeks.org/queue-using-stacks/)
2. [Design and Implement Special Stack Data Structure | Added Space Optimized Version](https://www.geeksforgeeks.org/design-and-implement-special-stack-data-structure/)
3. [Implement two stacks in an array](https://www.geeksforgeeks.org/implement-two-stacks-in-an-array/)
4. [Implement Stack using Queues](https://www.geeksforgeeks.org/implement-stack-using-queue/)
5. [Design a stack with operations on middle element](https://www.geeksforgeeks.org/design-a-stack-with-find-middle-operation/)
6. [How to efficiently implement k stacks in a single array?](https://www.geeksforgeeks.org/efficiently-implement-k-stacks-single-array/)
7. [How to create mergable stack?](https://www.geeksforgeeks.org/create-mergable-stack/)
8. [Design a stack that supports getMin() in O(1) time and O(1) extra space](https://www.geeksforgeeks.org/design-a-stack-that-supports-getmin-in-o1-time-and-o1-extra-space/)
9. [Implement a stack using single queue](https://www.geeksforgeeks.org/implement-a-stack-using-single-queue/)
10. [How to implement stack using priority queue or heap?](https://www.geeksforgeeks.org/implement-stack-using-priority-queue-or-heap/)
11. [Create a customized data structure which evaluates functions in O(1)](https://www.geeksforgeeks.org/create-customized-data-structure-evaluates-functions-o1/)

**Standard Problems based on Stack**

1. [Infix to Postfix Conversion using Stack](https://www.geeksforgeeks.org/stack-set-2-infix-to-postfix/)
2. [The Stock Span Problem](https://www.geeksforgeeks.org/the-stock-span-problem/)
3. [Check for balanced parentheses in an expression](https://www.geeksforgeeks.org/check-for-balanced-parentheses-in-an-expression/)
4. [Next Greater Element](https://www.geeksforgeeks.org/next-greater-element/)
5. [The Celebrity Problem](https://www.geeksforgeeks.org/the-celebrity-problem/)
6. [Expression Evaluation](https://www.geeksforgeeks.org/expression-evaluation/)
7. [Evaluation of Postfix Expression](https://www.geeksforgeeks.org/stack-set-4-evaluation-postfix-expression/)
8. [Iterative Tower of Hanoi](https://www.geeksforgeeks.org/iterative-tower-of-hanoi/)

**Operations on Stack**

1. [Reverse a stack using recursion](https://www.geeksforgeeks.org/reverse-a-stack-using-recursion/)
2. [Sort a stack using recursion](https://www.geeksforgeeks.org/sort-a-stack-using-recursion/)
3. [Sort a stack using a temporary stack](https://www.geeksforgeeks.org/sort-stack-using-temporary-stack/)
4. [Reverse a stack without using extra space in O(n)](https://www.geeksforgeeks.org/reverse-stack-without-using-extra-space/)

**Misc :**

1. [Iterative Postorder Traversal | Set 1 (Using Two Stacks)](https://www.geeksforgeeks.org/iterative-postorder-traversal/)
2. [Iterative Postorder Traversal | Set 2 (Using One Stack)](https://www.geeksforgeeks.org/iterative-postorder-traversal-using-stack/)
3. [Merge Overlapping Intervals](https://www.geeksforgeeks.org/merging-intervals/)
4. [Largest Rectangular Area in a Histogram | Set 2](https://www.geeksforgeeks.org/largest-rectangle-under-histogram/)
5. [Print ancestors of a given binary tree node without recursion](https://www.geeksforgeeks.org/print-ancestors-of-a-given-binary-tree-node-without-recursion/)
6. [Reverse a string using stack](https://www.geeksforgeeks.org/stack-set-3-reverse-string-using-stack/)
7. [Program for Tower of Hanoi](https://www.geeksforgeeks.org/c-program-for-tower-of-hanoi/)
8. [Find maximum depth of nested parenthesis in a string](https://www.geeksforgeeks.org/find-maximum-depth-nested-parenthesis-string/)
9. [Find maximum of minimum for every window size in a given array](https://www.geeksforgeeks.org/find-the-maximum-of-minimums-for-every-window-size-in-a-given-array/)
10. [Length of the longest valid substring](https://www.geeksforgeeks.org/length-of-the-longest-valid-substring/)
11. [Iterative Depth First Traversal of Graph](https://www.geeksforgeeks.org/iterative-depth-first-traversal/)
12. [Minimum number of bracket reversals needed to make an expression balanced](https://www.geeksforgeeks.org/minimum-number-of-bracket-reversals-needed-to-make-an-expression-balanced/)
13. [Check if a given array can represent Preorder Traversal of Binary Search Tree](https://www.geeksforgeeks.org/check-if-a-given-array-can-represent-preorder-traversal-of-binary-search-tree/)
14. [Form minimum number from given sequence](https://www.geeksforgeeks.org/form-minimum-number-from-given-sequence/)
15. [Find if an expression has duplicate parenthesis or not](https://www.geeksforgeeks.org/find-expression-duplicate-parenthesis-not/)
16. [Find maximum difference between nearest left and right smaller elements](https://www.geeksforgeeks.org/find-maximum-difference-between-nearest-left-and-right-smaller-elements/)
17. [Find next Smaller of next Greater in an array](https://www.geeksforgeeks.org/find-next-smaller-next-greater-array/)
18. [Find maximum sum possible equal sum of three stacks](https://www.geeksforgeeks.org/find-maximum-sum-possible-equal-sum-three-stacks/)
19. [Count natural numbers whose all permutation are greater than that number](https://www.geeksforgeeks.org/count-natural-numbers-whose-permutation-greater-number/)
20. [Delete consecutive same words in a sequence](https://www.geeksforgeeks.org/delete-consecutive-words-sequence/)
21. [Decode a string recursively encoded as count followed by substring](https://www.geeksforgeeks.org/decode-string-recursively-encoded-count-followed-substring/)
22. [Bubble sort using two Stacks](https://www.geeksforgeeks.org/bubble-sort-using-two-stacks/)
23. [Pattern Occurrences : Stack Implementation Java](https://www.geeksforgeeks.org/pattern-occurrences-stack-implementation-java/)
24. [Iterative method to find ancestors of a given binary tree](https://www.geeksforgeeks.org/iterative-method-to-find-ancestors-of-a-given-binary-tree/)
25. [Stack Permutations (Check if an array is stack permutation of other)](https://www.geeksforgeeks.org/stack-permutations-check-if-an-array-is-stack-permutation-of-other/)
26. [Tracking current Maximum Element in a Stack](https://www.geeksforgeeks.org/tracking-current-maximum-element-in-a-stack/)
27. [Check mirror in n-ary tree](https://www.geeksforgeeks.org/check-mirror-n-ary-tree/)
28. [Reverse a number using stack](https://www.geeksforgeeks.org/reverse-number-using-stack/)
29. [Reversing the first K elements of a Queue](https://www.geeksforgeeks.org/reversing-first-k-elements-queue/)
30. [Reversing a Queue](https://www.geeksforgeeks.org/reversing-a-queue/)
31. [Check if stack elements are pairwise consecutive](https://www.geeksforgeeks.org/check-if-stack-elements-are-pairwise-consecutive/)
32. [Spaghetti Stack](https://www.geeksforgeeks.org/g-fact-87/)
33. [Interleave the first half of the queue with second half](https://www.geeksforgeeks.org/interleave-first-half-queue-second-half/)
34. [Remove brackets from an algebraic string containing + and – operators](https://www.geeksforgeeks.org/remove-brackets-algebraic-string-containing-operators/)