

count digits of a number (2 approaches)
<u>reverse a number</u>
<u>inverse of a number</u>
<u>rotate a number</u>
gcd and lcm (in time compelxity $O(\min(a,b))$)
<u>gcd and lcm (optimized using euclid division algorithm)</u>
<u>prime factorization</u>
<u>bulb switcher</u>
Pattern 1
Pattern 2
Pattern 3
Pattern 4
Pattern 5
Pattern 6
Pattern 7
Pattern 8
Pattern 9
Pattern 10
Pattern 11
Pattern 12
Pattern 13
Pattern 14
Pattern 15

Pattern 16

Pattern 17

Pattern 18

Pattern 19

Pattern 20

decimal to anybase

any base to decimal

any base to any base

anybase additon

anybase subtraction

anybase multiplication

set a bit at a given position

unset a bit at the given position

toggle a bit at a given position

check whether the given bit set or not

Number of 1 Bits

Single number

number of 1 bits (kernighan)

number of 1 bits (kernighan aproach 2)

number of 1 bits (inbuilt function)

reverse integer

power of two

power of four

complement of base 10 integer

intro to pointers and its types

call by value and call by reference

pointer arithmetic

pointer to pointer

stack vs heap

dynamic memory allocation in c and cpp

1D array creation

2D array creation

References and its types (lvalue and rvalue)

Smart pointers

show call by value

show call by reference using pointers

show call by reference using references

Program of 1D array in stack

Program of 1D array in heap

Program of 2D array in heap (row-major order)

Program of 2D array in heap (using array of pointers)

analysis of time and space complexity

perform these operations

(cpp : stl vector and array two files) (java : arraylist ,one file)

insertion at end (push_back)(add) -> O(1)

removal at end (pop_back)(remove or removeLast) -> O(1)

insertion at random position (insert)(add(index)) ->O(n)

removal at random position (erase)(remove(index)) -> O(n)

find (find)(find and indexOf) -> O(n)

reverse(reverse) (collections.reverse) -> O(n)

min and max (min, max) (collections.min, collections.max) -> O(n)

fixed size array stl functions(swap,fill,sum,max,min)

Array using Structs

Array using Classes

Vector implementation {at().size.capacity.print.push_back,pop_back,insert erase, accumulate,min,max,reverse,find,operator overloading(==,=,[])}

Reference code

Sum of two arrays

Cyclically rotate array by one

Rotate array by k

"Second Largest in Array"

Reverse Array

Single Number 2

Next Permutation

Permutation Sequence

Factorial of large numbers

Move all negative elements to end (brute + optimized)

Sort Colors

Union of two arrays

Two Sum

container with most water

smallest Range I

Minimize the Heights I

minimize the Heights II

3 Sum

Merge Sorted Array without Extra[]

Majority Element

Majority Element II

3 Sum Closest

3Sum with Multiplicity

sqrt(x)

Valid Perfect Square

Peak index in a mountain range

First bad version

Koko Eating Bananas

Capacity To Ship Packages Within D Days

find the first and last position of element in sorted array

Find Peak Element

Search in Rotated Sorted Array

Aggresive Cows

Allocate books

Search in a 2d matrix

search in a 2d matrix 2

Cycle Sort

Set Mismatch

Find Duplicates in Array

Missing Number

Find all numbers disappeared in array

Find Duplicate Number

First missing positive

Custom Comparator

Largest Number

Custom Sort String

Shell sort

Merge Two Sorted Arrays

Pivot Index

Trapping rain water

Subarray sum equals K

Count Triplets That Can Form Two Arrays of Equal XOR

Range Sum Query 2D - Immutable

Max Sum Subarray of size K

Count Anagrams

Maximum No of Vowels

Longest Substring Without Repetition

Longest Substring Dynamic

Maximum Consecutive Ones III

Array traversal

1 to N

N to 1

Fibonacci

Binary search

Factorial

pow(x,n)

Count Good numbers

Print good numbers

Subset 1

Subset 2

Combination Sum 1

Generate all Binary strings(gfg)

Subset sum(gfg)

Letter Combination

Combination Sum 2

Combination Sum 3

Target Sum

Rat in a MAZE

N - Queens

Valid Sudoku

Sudoku Solver

Knights's Tour

Cherry Pickup

Difference of GCDs

GCD Partition

Common Divisors

Exponentiation

Design Linked List

Reverse Linked List(iterative and recurrision)

Palindrome(Brte and Optimized)

HOMEWORK

Reverse Linked List 2

Add Two numbers

Add Two numbers 2

Design Linked List 2 - LeetCode

Reverse a Doubly Linked List

Detect cycle

Flatten a Multilevel Doubly Linked List - LeetCode

HOMEWORK

fibonacci number

N-th Tribonacci Number

Climbing Stairs

Count ways to N'th Stair

Jump Game

HOMEWORK

Dice Combinations

Minimizing Coins

Coin Combinations I

Duplicate Number

Intersection of two linked lists

clone list with random pointer

HOMEWORK

Insertion sort list

merge sort list

Count inversions

LRU Cache

LFU Cache

preorder traversal

post order traversal

HOMEWORK

max and min element in binary tree

sum of binary tree

Invert Binary Tree

Maximum Depth

Path sum

Path sum II

Path Sum III

N-Arry Post Order Traversal

Path sum III (OPTIMISED)

level order traversal

level order traversal II

zigzag level order traversal

diameter of binary tree

LCA of binary tree

left side binary tree (both DFS and BFS)

right side view binary tree(both DFS and BFS)

balanced binary tree

Same tree

Boundary Traversal

Top view of binary tree

Bottom view of binary tree

Vertical order traversal

Binary Tree Cameras

Distribute Coins in Binary tree

Delete Leaf Node

Delete leaves with given value

Delete Nodes and return forest

Binary Tree Paths

Find Bottom left tree value

Pseudo palindromic paths in a binary tree

Serialize and Deserialize Binary Tree

All Nodes Distance K in Binary tree (BFS)

Maximum Path Sum

All Nodes Distance K in Binary tree (DFS)

K distance from root(BFS & DFS)

Binary tree to be infected (BFS & DFS)

Path to given node

Construct Binary Tree from Preorder and Inorder

Construct Binary tree from Inorder and Postorder

Duplicate SubTree

Maximum Width of Binary Tree

preorder traversal (using stack)

inorder traversal(using stack)

post order traversal (using stack)

Implement Trie

Word Break

Shortest Unique prefix

minimum xor pair

Populating next right pointers(bfs)

Populating next right pointers(dfs)

Populating next right pointers(iterative)

Populating next right pointers 2(dfs)

Insert in binary search tree

search in binary search tree

Minimum element in binary search tree

ceil in binary search tree

floor in binary search tree

delete node in binary search tree

validate binary search tree

binary search tree to greater sum tree

maximum sum bst in binary tree