Contents

[Basic contest 2](#_Toc144946940)

[Basics 2](#_Toc144946941)

[Brute Force 2](#_Toc144946942)

[selection statement 2](#_Toc144946943)

[loops 2](#_Toc144946944)

[string 3](#_Toc144946945)

[math 4](#_Toc144946946)

[array 5](#_Toc144946947)

[matrix (2d array) 5](#_Toc144946948)

[vector (1D & 2D) 5](#_Toc144946949)

[sort 6](#_Toc144946950)

[pair 6](#_Toc144946951)

[Function 6](#_Toc144946952)

[Primes 7](#_Toc144946953)

[Map 7](#_Toc144946954)

[set 8](#_Toc144946955)

[dp 8](#_Toc144946956)

[Bitmask 8](#_Toc144946957)

[Two pointers 8](#_Toc144946958)

[Sliding Window 9](#_Toc144946959)

[Dequeue 9](#_Toc144946960)

[Queue 9](#_Toc144946961)

[Binary search 9](#_Toc144946962)

[Kadane’s Algorithm (maximum sum subarray) 10](#_Toc144946963)

[prefix – suffix 10](#_Toc144946964)

[Linked List 10](#_Toc144946965)

[Aggregation 11](#_Toc144946966)

[Geometry 11](#_Toc144946967)

[Graph 11](#_Toc144946968)

[Stack 11](#_Toc144946969)

[Multi set 11](#_Toc144946970)

[Tree 12](#_Toc144946971)

[uint32\_t 12](#_Toc144946972)

[max element 12](#_Toc144946973)

[Binary Search Tree 12](#_Toc144946974)

[Greedy 12](#_Toc144946975)

# Basic contest

<https://codeforces.com/group/ANr5IuAgTk/contest/217491>

# Basics

<https://www.hackerrank.com/challenges/cpp-hello-world/problem?isFullScreen=true>

<https://www.hackerrank.com/challenges/cpp-input-and-output/problem?isFullScreen=true>

# Brute Force

<https://codeforces.com/problemset/problem/922/B> \*\*Good for XOR (1300)

<https://codeforces.com/problemset/problem/919/B> \*\*Recursion (1100)

<https://codeforces.com/problemset/problem/146/B> (1300)

<https://codeforces.com/problemset/problem/911/B> (1200)

<https://codeforces.com/problemset/problem/907/A> (1300)

# selection statement

<https://www.hackerrank.com/challenges/c-tutorial-conditional-if-else/problem?isFullScreen=true>

<https://codeforces.com/problemset/problem/4/A> \*\*edge cases ----(Qualification sheet) (800)

<https://codeforces.com/problemset/problem/546/A> \*\*max func., n\*(n+1)/2 \*\*extra week 1 (800)

<https://codeforces.com/problemset/problem/122/A> \*\*importance of constrains (1000)

<https://codeforces.com/problemset/problem/479/A> \*\*not recommended

<https://codeforces.com/problemset/problem/466/A> \*\*a good example on min function (1200)

<https://codeforces.com/problemset/problem/1/A> (1000)

# loops

<https://codeforces.com/problemset/problem/1351/A> \*\*testcases (800)

<https://codeforces.com/problemset/problem/69/A> ----(week1 - level1) (1000)

<https://codeforces.com/problemset/problem/791/A> ----(Qualification round&sheet) (800)

<https://codeforces.com/problemset/problem/705/A> \*\*flag idea ----(week1 - level1) (800)

<https://codeforces.com/problemset/problem/977/A> (800)

<https://codeforces.com/problemset/problem/148/A> (800)

<https://codeforces.com/contest/734/problem/A> (800)

<https://codeforces.com/problemset/problem/116/A> (800)

<https://codeforces.com/problemset/problem/271/A> (800)

<https://codeforces.com/problemset/problem/677/A> ----(week1 - level1) (800)

<https://codeforces.com/problemset/problem/467/A> (800)

<https://codeforces.com/problemset/problem/344/A> (800)

<https://codeforces.com/problemset/problem/1030/A> (800)

<https://codeforces.com/problemset/problem/144/A> (800)

<https://codeforces.com/problemset/problem/580/A> \*\*max func. “recommended” (900)

<https://codeforces.com/problemset/problem/510/A> (800)

<https://codeforces.com/problemset/problem/158/B> \*\*taxi - recommended (1100)

<https://codeforces.com/problemset/problem/1729/A> \*\*abs function (800)

<https://codeforces.com/contest/1690/problem/A> (800)

<https://codeforces.com/problemset/problem/1506/A> (800)

<https://codeforces.com/problemset/problem/1790/A> \*\*good (800)

<https://atcoder.jp/contests/abc275/tasks/abc275_a> \*\*recommended for the qualification contest

<https://codeforces.com/problemset/problem/1778/A> (800)

<https://codeforces.com/problemset/problem/1777/A> (800)

<https://www.hackerrank.com/challenges/c-tutorial-for-loop/problem?isFullScreen=true>

# string

<https://atcoder.jp/contests/abc279/tasks/abc279_a> \*\*easy

<https://codeforces.com/problemset/problem/785/A> (800)

<https://codeforces.com/problemset/problem/71/A> (800)

<https://codeforces.com/problemset/problem/61/A> \*\*short hand if else (800)

<https://codeforces.com/problemset/problem/266/B> (800)

<https://codeforces.com/problemset/problem/208/A> \*\*recommended (900)

<https://codeforces.com/problemset/problem/282/A> (800)

<https://codeforces.com/problemset/problem/118/A> (1000)

<https://codeforces.com/problemset/problem/112/A> (800)

<https://codeforces.com/problemset/problem/339/A> (800)

<https://codeforces.com/problemset/problem/281/A> (800)

<https://codeforces.com/problemset/problem/266/A> (800)

<https://codeforces.com/problemset/problem/236/A> (800) \*\*frequency array

<https://codeforces.com/problemset/problem/96/A> (900)

<https://codeforces.com/problemset/problem/59/A> (800)

<https://codeforces.com/problemset/problem/58/A> (1000)

<https://codeforces.com/problemset/problem/41/A> (800)

<https://codeforces.com/problemset/problem/734/A> (800)

<https://codeforces.com/problemset/problem/133/A> (900)

<https://codeforces.com/problemset/problem/141/A> \*\*frequency array (800)

<https://codeforces.com/problemset/problem/520/A> \*\*transform + all\_of functions, tolower (800)

<https://codeforces.com/problemset/problem/1722/A> (800)

<https://codeforces.com/problemset/problem/1722/B> (800)

<https://codeforces.com/problemset/problem/1791/A> (800)

<https://codeforces.com/problemset/problem/1772/A> (800)

<https://codeforces.com/problemset/problem/1791/B> (800)

<https://codeforces.com/problemset/problem/131/A> (1000)

<https://codeforces.com/problemset/problem/443/A> \*\*getline example, frequency array (800)

<https://atcoder.jp/contests/abc276/tasks/abc276_a> \*\*rfind function

<https://codeforces.com/problemset/problem/1800/A>

<https://codeforces.com/problemset/problem/1790/B>

<https://codeforces.com/problemset/problem/1795/A>

<https://codeforces.com/problemset/problem/1800/D>

<https://codeforces.com/problemset/problem/1774/A>

<https://codeforces.com/problemset/problem/1796/B> \*\*hard

<https://leetcode.com/problems/reverse-words-in-a-string-iii/>

<https://leetcode.com/problems/number-of-lines-to-write-string/> \*\*like frequency array

<https://codeforces.com/contest/550/problem/A>

# math

<https://codeforces.com/problemset/problem/50/A> (800)

<https://codeforces.com/problemset/problem/200/B> (800)

<https://codeforces.com/problemset/problem/617/A> (800)

<https://codeforces.com/problemset/problem/318/A> \*\*TLE error (900)

<https://codeforces.com/problemset/problem/486/A> \*\*TLE error \*\* complexity (800)

<https://codeforces.com/problemset/problem/1328/A> (800)

<https://codeforces.com/problemset/problem/996/A> (800)

<https://codeforces.com/problemset/problem/1335/A> \*\*TLE error (800)

<https://codeforces.com/problemset/problem/1343/A> \*\*unsigned long long (900)

<https://codeforces.com/problemset/problem/742/A> \*\*pattern of power (1000) \*\*guzel

<https://codeforces.com/problemset/problem/1535/A> \*\*max function (800)

<https://codeforces.com/contest/1311/problem/A> (800)

<https://codeforces.com/problemset/problem/1766/A> \*\*log10 (800)

<https://leetcode.com/problems/subtract-the-product-and-sum-of-digits-of-an-integer/>

<https://codeforces.com/problemset/problem/1389/A> (800)

<https://codeforces.com/group/ANr5IuAgTk/contest/217629/problem/E> \*\*XOR

<https://codeforces.com/problemset/problem/230/B> prime numbers (1300)

<https://codeforces.com/problemset/problem/520/B> (1400)

# array

<https://codeforces.com/problemset/problem/231/A> -- used in GDSC Qualification (800)

<https://codeforces.com/problemset/problem/168/A> (900)

<https://codeforces.com/problemset/problem/136/A> (800)

<https://codeforces.com/problemset/problem/996/A> (800)

<https://codeforces.com/problemset/problem/337/A> (900)

<https://atcoder.jp/contests/abc284/tasks/abc284_a> \*\*array of strings \*\*reverse

<https://www.hackerrank.com/challenges/arrays-introduction/problem?isFullScreen=true>

<https://www.hackerrank.com/challenges/variable-sized-arrays/problem?isFullScreen=true>

<https://atcoder.jp/contests/abc278/tasks/abc278_a>

# matrix (2d array)

<https://codeforces.com/problemset/problem/263/A> (800)

# vector (1D & 2D)

<https://leetcode.com/problems/rotate-array/?envType=study-plan&id=algorithm-i> \*\*reverse function

<https://leetcode.com/problems/squares-of-a-sorted-array/> \*\*sort function

<https://leetcode.com/problems/search-a-2d-matrix/> \*\*2D

<https://leetcode.com/problems/valid-sudoku/> \*\*2D

<https://leetcode.com/problems/reshape-the-matrix/> \*\*2D

<https://codeforces.com/contest/803/problem/B>

<https://codeforces.com/contest/998/problem/B>

<https://codeforces.com/problemset/problem/1352/A>

<https://codeforces.com/problemset/problem/25/A>

<https://codeforces.com/problemset/problem/1742/A>

<https://codeforces.com/problemset/problem/1794/A>

[**https://codeforces.com/problemset/problem/1788/A**](https://codeforces.com/problemset/problem/1788/A)

[**https://codeforces.com/problemset/problem/1780/A**](https://codeforces.com/problemset/problem/1780/A)

[**https://leetcode.com/problems/find-nearest-point-that-has-the-same-x-or-y-coordinate/**](https://leetcode.com/problems/find-nearest-point-that-has-the-same-x-or-y-coordinate/)

[**https://leetcode.com/problems/move-zeroes/**](https://leetcode.com/problems/move-zeroes/)

# sort

<https://codeforces.com/problemset/problem/230/A> \*\*pair

<https://codeforces.com/problemset/problem/160/A>

<https://codeforces.com/problemset/problem/1792/A>

<https://codeforces.com/problemset/problem/405/A> \*\*foreach

<https://codeforces.com/problemset/problem/1767/B>

<https://leetcode.com/problems/largest-perimeter-triangle/>

<https://cses.fi/problemset/task/1619/>

<https://cses.fi/problemset/task/1640/> \*\*lower bound on pairs

# pair

<https://codeforces.com/problemset/problem/230/A>

<https://codeforces.com/problemset/problem/268/A>

# Function

<https://codeforces.com/problemset/problem/110/A>

<https://atcoder.jp/contests/abc273/tasks/abc273_a>

<https://atcoder.jp/contests/abc283/tasks/abc283_a> \*\*power function

<https://codeforces.com/problemset/problem/822/A> \*\*factorial (800)

<https://codeforces.com/contest/1244/problem/A> \*\*ceil (800)

<https://codeforces.com/problemset/problem/476/A> \*\*ceil (1000)

<https://codeforces.com/problemset/problem/1792/B> (1200) min & max functions

<https://codeforces.com/problemset/problem/1850/E> \*\*or binary search (1100)

# Primes

<https://codeforces.com/problemset/problem/26/A> \*\*prime factors (900)

# Map

<https://codeforces.com/problemset/problem/785/A>

<https://codeforces.com/problemset/problem/4/C>

<https://codeforces.com/problemset/problem/443/A>

<https://leetcode.com/problems/isomorphic-strings/> \*\*round 2

<https://leetcode.com/problems/longest-palindrome/?envType=study-plan&id=level-1>

<https://codeforces.com/problemset/problem/1520/D> \*\*recommended

<https://codeforces.com/problemset/problem/1742/B>

<https://codeforces.com/problemset/problem/1800/B>

<https://codeforces.com/problemset/problem/1791/D>

<https://codeforces.com/problemset/problem/1790/D>

<https://codeforces.com/problemset/problem/1771/A>

<https://practice.geeksforgeeks.org/problems/find-the-frequency/1>

<https://practice.geeksforgeeks.org/problems/twice-counter4236/1>

<https://leetcode.com/problems/two-sum-ii-input-array-is-sorted/>

<https://leetcode.com/problems/intersection-of-two-arrays-ii/>

<https://codeforces.com/problemset/problem/977/B>

<https://codeforces.com/problemset/problem/204/B> \*\* map of pair

# set

<https://cses.fi/problemset/task/1621/>

<https://codeforces.com/problemset/problem/228/A>

<https://codeforces.com/problemset/problem/469/A>

<https://codeforces.com/problemset/problem/1512/A> \*\*set & map

<https://codeforces.com/problemset/problem/22/A>

[https://leetcode.com/problems/contains-duplicate/](https://leetcode.com/problems/contains-duplicate/?envType=study-plan&id=data-structure-i)

[https://leetcode.com/problems/two-sum/description/](https://leetcode.com/problems/two-sum/description/?envType=study-plan&id=data-structure-i)

[**https://codeforces.com/problemset/problem/1790/C**](https://codeforces.com/problemset/problem/1790/C) **\*\*hard**

[**https://codeforces.com/problemset/problem/1203/E**](https://codeforces.com/problemset/problem/1203/E) **\*\*recommended**

[**https://leetcode.com/problems/kth-missing-positive-number/**](https://leetcode.com/problems/kth-missing-positive-number/)

# dp

<https://codeforces.com/contest/189/problem/A>

<https://leetcode.com/problems/climbing-stairs/>

<https://leetcode.com/problems/fibonacci-number/>

<https://leetcode.com/problems/n-th-tribonacci-number/>

<https://practice.geeksforgeeks.org/problems/path-in-matrix3805/1>

<https://leetcode.com/problems/jump-game/>

<https://leetcode.com/problems/min-cost-climbing-stairs/>

<https://codeforces.com/contest/1829/problem/G> \*\*good for the contest

<https://codeforces.com/problemset/problem/1851/E> \*\*good

# Bitmask

<https://codeforces.com/problemset/problem/550/B>

<https://codeforces.com/problemset/problem/1790/E> \*\*hard

<https://codeforces.com/problemset/problem/1230/A>

# Two pointers

<https://codeforces.com/problemset/problem/1791/C>

<https://codeforces.com/problemst/problem/1462/A>

<https://leetcode.com/problems/is-subsequence/> \*\*subsequence

<https://leetcode.com/problems/remove-element/>

[https://leetcode.com/problems/merge-sorted-array/](https://leetcode.com/problems/merge-sorted-array/?envType=study-plan&id=data-structure-i)

<https://leetcode.com/problems/remove-duplicates-from-sorted-array/>

<https://leetcode.com/problems/remove-duplicates-from-sorted-array-ii/>

<https://codeforces.com/problemset/problem/1676/F>

<https://codeforces.com/contest/372/problem/A>

<https://codeforces.com/problemset/problem/381/A>

# Sliding Window

<https://codeforces.com/problemset/problem/279/B> \*\*repeated

# Dequeue

<https://codeforces.com/problemset/problem/1729/B>

<https://codeforces.com/problemset/problem/1511/C>

<https://codeforces.com/problemset/problem/381/A>

# Queue

<https://codefomrces.com/problemset/problem/279/B>

<https://codeforces.com/problemset/problem/1579/E1>

# Binary search

[https://leetcode.com/problems/binary-search/](https://leetcode.com/problems/binary-search/?envType=study-plan&id=algorithm-i)

[https://leetcode.com/problems/first-bad-version/](https://leetcode.com/problems/first-bad-version/?envType=study-plan&id=algorithm-i)

[https://leetcode.com/problems/search-insert-position/](https://leetcode.com/problems/search-insert-position/?envType=study-plan&id=algorithm-i)

<https://codeforces.com/problemset/problem/1791/F> \*\*lower bound

<https://leetcode.com/problems/koko-eating-bananas/>

<https://leetcode.com/problems/guess-number-higher-or-lower/>

<https://codeforces.com/gym/101021/problem/1> \*\*interactive problem

<https://leetcode.com/problems/valid-perfect-square/>

<https://leetcode.com/problems/sqrtx/>

<https://leetcode.com/problems/peak-index-in-a-mountain-array/>

<https://leetcode.com/problems/find-smallest-letter-greater-than-target/> \*\*upper bound

<https://leetcode.com/problems/arranging-coins/>

<https://leetcode.com/problems/count-negative-numbers-in-a-sorted-matrix/>

<https://leetcode.com/problems/special-array-with-x-elements-greater-than-or-equal-x/> \*\*lower bound

<https://codeforces.com/problemset/problem/1676/E> \*\*lower bound

<https://codeforces.com/contest/1324/problem/D> \*\* upper bound

<https://codeforces.com/problemset/p8roblem/1850/E> \*\*or math

<https://cses.fi/problemset/task/1091/> \*\*multiset upper bound

# Kadane’s Algorithm (maximum sum subarray)

[https://leetcode.com/problems/maximum-subarray/](https://leetcode.com/problems/maximum-subarray/?envType=study-plan&id=data-structure-i)

<https://cses.fi/problemset/task/1643>

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock/description/?envType=study-plan&id=level-1>

# prefix – suffix

<https://leetcode.com/problems/running-sum-of-1d-array/>

<https://leetcode.com/problems/find-pivot-index/>

<https://codeforces.com/problemset/problem/545/D> \*\*sort

<https://leetcode.com/problems/average-salary-excluding-the-minimum-and-maximum-salary/>

<https://codeforces.com/contest/1697/problem/B>

# Linked List

<https://leetcode.com/problems/reverse-linked-list/>

<https://leetcode.com/problems/merge-two-sorted-lists/>

<https://leetcode.com/problems/split-linked-list-in-parts/>

<https://leetcode.com/problems/remove-duplicates-from-sorted-list/>

<https://leetcode.com/problems/add-two-numbers/>

<https://leetcode.com/problems/middle-of-the-linked-list/>

<https://leetcode.com/problems/linked-list-cycle-ii/>

<https://leetcode.com/problems/remove-nth-node-from-end-of-list/>

<https://leetcode.com/problems/linked-list-cycle/description/>

<https://leetcode.com/problems/intersection-of-two-linked-lists/>

<https://leetcode.com/problems/remove-linked-list-elements/>

<https://leetcode.com/problems/palindrome-linked-list/>

# Aggregation

<https://leetcode.com/problems/sum-of-two-integers/>

# Geometry

<https://codeforces.com/problemset/problem/650/A> (1400)

# Graph

<https://leetcode.com/problems/maximum-depth-of-binary-tree/> (dfs)

<https://leetcode.com/problems/flood-fill/> (dfs)

<https://leetcode.com/problems/number-of-islands/> (dfs)

<https://leetcode.com/problems/max-area-of-island/> (dfs)

<https://codeforces.com/problemset/problem/580/C> (dfs) 1500

# Stack

<https://leetcode.com/problems/valid-parentheses/>

<https://leetcode.com/problems/backspace-string-compare/>

<https://codeforces.com/problemset/problem/1374/C>

<https://codeforces.com/problemset/problem/1452/C>

# Multi set

<https://codeforces.com/problemset/problem/1800/C1>

<https://codeforces.com/problemset/problem/1800/C2>

<https://codeforces.com/problemset/problem/1791/G1>

<https://cses.fi/problemset/task/1091/> \*\*upper bound

# Tree

<https://leetcode.com/problems/n-ary-tree-preorder-traversal/>

<https://leetcode.com/problems/binary-tree-level-order-traversal/>

# uint32\_t

<https://leetcode.com/problems/number-of-1-bits/description/>

# max element

<https://leetcode.com/problems/peak-index-in-a-mountain-array/>

# Binary Search Tree

<https://leetcode.com/problems/convert-sorted-array-to-binary-search-tree/>

<https://leetcode.com/problems/find-mode-in-binary-search-tree/description/>

<https://leetcode.com/problems/validate-binary-search-tree/>

# Greedy

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/>