

COMPETITIVE PROGRAMMING

COVID CODING PROGRAM

MENTOR: MR. RAHUL LATHER
sachinlather49@gmail.com

06 April 2020

What is Competitive Programming?

Competitive programming is a mind sport usually held over the Internet, involving participants trying to program according to provided specifications. Contestants are referred to as *sport programmers*. Competitive programming is recognized and supported by several multinational software and Internet companies, such as Google and Facebook. There are several organizations who host programming competitions on a regular basis([Codeforces](#), [Codechef](#), [TopCoder](#), [Google's Codejam](#), Facebook Hacker Cup etc).

Content To Be Discussed Today!

- Mathematical Fundamentals
 - Factorial, Power of a Number, Prime Numbers
 - Modulo Arithmetic.
 - BIT Manipulation.
- Arrays, Matrices

Mathematical Fundamentals

1. **Program :-** [Factorial of a small number](#)
2. **Program :-** [Power of a number\(X\) raised to another number\(Y\), Optimized Version](#)
3. **Program :-** [Print All Prime Numbers smaller than N , \(Optimized Version\)](#)

An Introduction to Modular Math -

When we divide two integers we will have an equation that looks like the following:

$$A / B = Q \text{ remainder } R,$$

A -> is the dividend, B -> is the divisor, Q -> is the quotient, R -> is the remainder.

The Modulo Operator(%) or (mod):

A % B gives the remainder R when A is divided B i.e $A \% B = R$.

Ex. $8 \% 3 = ?$, $8 \% 4 = ?$, maximum value of result $A \% B$?.

Modular addition and subtraction:

$$(A + B) \bmod C = (A \bmod C + B \bmod C) \bmod C$$

$$(A - B) \bmod C = (A \bmod C - B \bmod C) \bmod C$$

Multiplication property of modular arithmetic:

$$(A * B) \bmod C = (A \bmod C * B \bmod C) \bmod C$$

Modular exponentiation(Power in Modular Arithmetic):

Program :- [Given three numbers x, y and p, compute \$\(x^y\) \% p\$](#)

(First calculating power and then taking modulo can give wrong results due to overflow).

BIT MANIPULATION:

Bitwise Operators:

1. The **& (bitwise AND)** in C or C++ takes two numbers as operands and does AND on every bit of two numbers. The result of AND is 1 only if both bits are 1.
2. The **| (bitwise OR)** in C or C++ takes two numbers as operands and does OR on every bit of two numbers. The result of OR is 1 if any of the two bits is 1.
3. The **^ (bitwise XOR)** in C or C++ takes two numbers as operands and does XOR on every bit of two numbers. The result of XOR is 1 if the two bits are different.
4. The **<< (left shift)** in C or C++ takes two numbers, left shifts the bits of the first operand, the second operand decides the number of places to shift.
5. The **>> (right shift)** in C or C++ takes two numbers, right shifts the bits of the first operand, the second operand decides the number of places to shift.
6. The **~ (bitwise NOT)** in C or C++ takes one number and inverts all bits of it

Bitwise Hacks for Competitive Programming:

1. [Bit Tricks for Competitive Programming](#)
2. [Count set bits in integer\(Brian Kernighan's Algorithm\)](#)
3. [Checking if given 32 bit integer is power of 2](#)

Arrays & Matrices(Multidimensional Arrays):

An array is a collection of similar type of items stored at contiguous memory locations.

Whereas, [multidimensional arrays](#) in simple words as array of arrays. Data in multidimensional arrays are stored in tabular form (in row major order).

Arrays STL :

Sorting an array,

Syntax: `sort(arr, arr + n);`

where arr is the array and n is the size of the array.

Size of array : `sizeof(arr) / sizeof(data_type of array)`

Searching in an array : [Linear Search](#), [Binary Search](#).

How to Approach a problem in Competitive Programming?

1. **Paragraphs :** Read the problem statement carefully 2-3 times unless it is clarified.
2. **Input Format :** Take Input in the format as explained in this section.
3. **Output Format :** Print Output in the format as explained in this section.
4. **Constraints :** Identifies Data Types and Accepted Time Complexity of Solution.
5. **Sample IO :** Explains the sample test cases for better explanation of the problem.

Analysis of Constraints :

Time Limit (Total Operations we can perform is 10^8 /sec).

Problem : <https://www.codechef.com/problems/LECANDY>

Problem(2D) : <https://www.geeksforgeeks.org/find-the-row-with-maximum-number-1s/>
(Amazon Interview Question)

