Greetings.

This document is based on my experience. I think this will be fruitful for others so I like to share it with everyone. My suggestion is that if your rating is <= 1100 or >= 1400 in codeforces you can skip it here.

Special thanks to MD. REJWAN KABIR HAMIM(Hamim99), ASHFAQUR RAHMAN ADIB(adibur6), and HASAN MESBAUL ALI TAHER for their valuable recommendations and reviews.

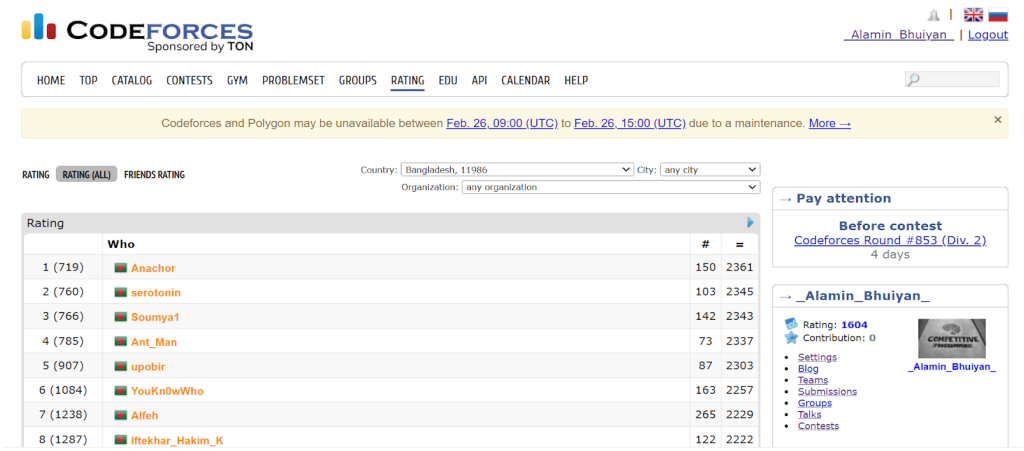
**Just a reminder: There is no shortcut in competitive programming. These strategies are just to improve the process of thinking that we have gained from our experiences.

Let's start!

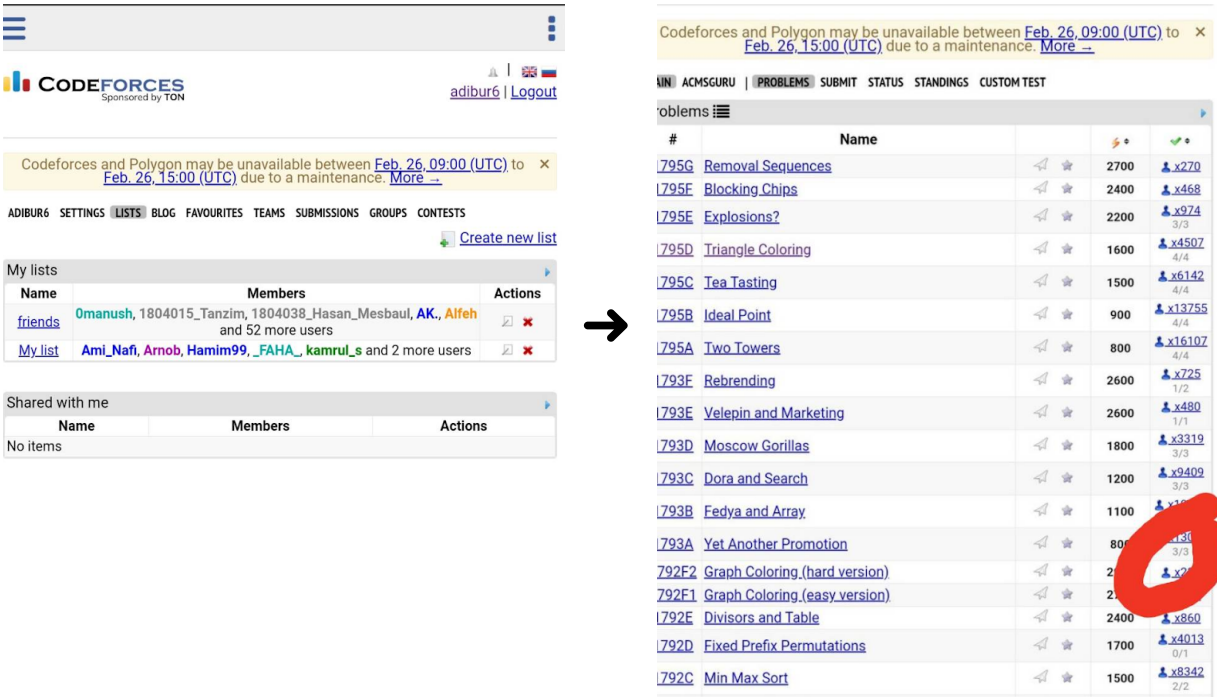
As you are reading this, I hope you know about basic programming languages and online judges. Here I will discuss how to think and how to solve problems. I have also added some problems as examples (the related problems can be solved in other ways too). Note that these problems are not enough to practice. These problems are provided here so that you can relate a problem with the tactics.

Pre-tips: Knowing Codeforces better

This section is suggested by Ashfaqur Rahman Adib(adibur6). Those who are eager for competitive programming need to love this sector and also try to find interesting content in competitive programming. Competitive programming is a long-term process, it will be better if you make it fascinating.
It is also suggested that you keep track of your day-to-day problem-solving.
Let's see codeforces at a glance:
In Codeforces you can see the list of Bangladeshi programmers in the Rating section.



Add them to your friend list and follow their hard work and dedication.
There is a catalog section where you can find a lot of resources.
There is also the idea of making a list. You will find an option named list in your profile. Here you can make a list that will help you to understand which problem is solved by your listed persons.



Now move on to problem-solving strategies.

While problem-solving, first I follow some strategies and then I try to solve it through categories. In this document, I will try to explain these two tactics.

Scenario

Suppose, you have a problem in front of you. What should we observe in this situation?

- Statement
- Constraints(*)
- Input/Output(*)
- Sample Case
- Notes

*Give more importance

Strategies

Now we have a problem. We need to understand what they want and what should be done. This is how I like to think.

Do not believe his lies

Never believe what the statement says at the first chance. You have an input section and an output section. What would be the easiest way you could solve this situation? The statement does not tell you the efficient way. Now, try with your own thoughts and easy ways.

Related Problems:

- *https://codeforces.com/contest/1605/problem/B*

Brute force

If you find no solutions to the given problem, can you solve it with loop execution? Try to run a loop to solve the given ideas. This idea is called brute force. Now solve the problem using brute force.

Then you should try to make the solution efficient, small, and faster. What do they mean?

- Idea 1 Explanation: Suppose your solution has 3 loops can you make it in 2?
  - Example:
    - You are given two integers K and S. Three variables X, Y, and Z take integer values satisfying 0≤X, Y, Z≤K. How many different assignments of values to X, Y, and Z are there such that X+Y+Z=S?
  - Below I am going to show you two solutions. I have commented on the procedures.
    - The second solution is more efficient, smaller, and faster than the first one.

Solution 1: using 3 Loops:

```cpp
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int k , s , x , y , z, cnt = 0;
    cin >> k >> s; //taking k and s as input
    for(x = 0; x <= k ; x++) //First loop
    {
        for(y = 0; y <= k; y++) //Second loop
        {
            for(z = 0; z <= k; z++) //Third loop
            {
                if(x+y+z == s) //checking x+y+z is equals to s or not
                {
                    cnt++; //increment count
                }
            }
        }
    }
    cout << cnt << endl; // print the output
}
```

Solution 2: using 2 loops:

```cpp
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int k , s , x , y , z, cnt = 0;
    cin >> k >> s; //taking k and s as input
    for(x = 0; x <= k ; x++) //First loop
    {
        for(y = 0; y <= k; y++) //Second loop
        {
            z = s-(x+y); // calculating the third value
            if(z >= 0 && z <= k) // checking if z is in the range
            {
                cnt++; // increment count
            }
        }
    }
    cout << cnt << endl; // print the output
}
```

- Idea 2 Explanation: Is there any formula that can eliminate a loop?
    - Example:
        - Given n. What is the sum of 1+2+3+.....+n?

Solution 1: Using Loop

```cpp
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int n, sum = 0;
    cin >> n; // taking input
    for(int i = 1; i <= n; i++)
    {
        sum = sum + i; // adding i to the sum
    }
    cout << sum << endl;
}
```

Solution 2: Using formula

```cpp
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int n, sum = 0;
    cin >> n; // taking input
    sum = (n*(n+1))/2; //formula of generating sum of 1 to n
    cout << sum << endl;
}
```

Also, you can try recursion. Sometimes there might be a solution idea that goes deeper and deeper. That idea can't be executed with loops. In this situation, recursion helps a lot.

To learn more about recursion we can see youtube and blogs online.

Resource: Recursion Introduction and Identification

This video can help a lot but this is not enough. I suggest you watch more videos on recursion and read more about recursion.

Related Problems:

- *https://atcoder.jp/contests/abc051/tasks/abc051_b*
- *https://codeforces.com/contest/1194/problem/C*
- *https://codeforces.com/contest/1221/problem/A*
- https://lightoj.com/problem/the-fastest-sorting-ever (if you are able to solve this one, you should be proud of your recursive skill. And do not forget to brag about writing a code that writes a code, like ChatGPT)

New topic suggestion: Backtracking with branch pruning

Constructive algorithms

Sometimes we need to observe logically. For example,

- There might be thousands of elements in an array but the middle one or the left one, or the right one is the answer.
- Can you see any pattern from the test cases?
- To find patterns, get your hands dirty. That is, manually generate some input-outputs, possibly consecutive

Related Problems:

- *https://codeforces.com/contest/1606/problem/A*
- *https://codeforces.com/contest/1205/problem/A*
- *https://codeforces.com/contest/1221/problem/B*
- *https://codeforces.com/contest/1425/problem/H*
- *https://codeforces.com/problemset/problem/1381/A1*
- Game with Chips

Corner Cases/Test Cases

At this point, we have solved the problem using our above strategies. But still, we are getting "Wrong Answer". Now we should generate more test cases by ourselves.

There are some tips to find more cases:

- Check if your answer is correct when n = 1, largest possible value, -1, 0, odd, even, prime, non-prime

- If you have only one number(n) as input, it is better if you check all the test cases manually from n = 1 to n = 20
- If the problem gives two numbers as input: n and m. There is a trick to check test case

n	m	Description
even	even	Take an even number for n and an even number for m
even	odd	Take an even number for n and an odd number for m
odd	even	Take an odd number for n and an even number for m
odd	odd	Take an odd number for n and an odd number for m

  - You know the pattern right?

Related Problems:

- *https://codeforces.com/contest/1184/problem/A1*(TC,20)
- *https://codeforces.com/contest/1610/problem/A*
- *https://codeforces.com/contest/1562/problem/C*

Corner Case:
00,10,01,11
111,111,100,110,010,000

- *https://codeforces.com/problemset/problem/887/A*(zero>=6)
- *https://codeforces.com/contest/1589/problem/B*
- *https://codeforces.com/contest/1582/problem/B*
- *https://codeforces.com/contest/1581/problem/B*
- *https://codeforces.com/contest/1256/problem/B*
- *https://www.codechef.com/LTIME98B/problems/ARRT*
- *https://codeforces.com/contest/1469/problem/C*
- *https://codeforces.com/contest/1598/problem/B*

Traverse from back

Sometimes it is better to traverse the array in reverse order. Your idea is okay, just travel it from the end.

Related Problems:

- *https://codeforces.com/contest/1560/problem/D*

Categories

Some problems have specific categories. To solve these types of problems it is better to know some procedures. These procedures are called tags. The procedures make the solution

optimized. Some of us can generate these ideas by ourselves during contest time but it is better to know these tags as some basic rules.

Math

Sometimes we can see some math-type problems. Instead of running a loop, it is better to use plus, minus, multiplication, division, modulus, square root, etc.

Related Problems:

- *https://codeforces.com/contest/1581/problem/A*
- *https://toph.co/p/ashik-s-trouble*
- *https://codeforces.com/contest/1613/problem/A*
- *https://codeforces.com/contest/1605/problem/A*
- *https://codeforces.com/contest/1589/problem/A*
- *https://codeforces.com/contest/1593/problem/B*
- *https://atcoder.jp/contests/abc195/tasks/abc195_b*
- *https://codeforces.com/contest/1607/problem/B*
- *https://codeforces.com/problemset/problem/1374/D*

Greedy

Maybe you need to find the minimum or maximum value. Just think more and more to solve it as easily as possible. To improve on greedy ideas you need to solve a lot of problems. It is one of the most important techniques to solve a problem

Related Problems:

- *https://codeforces.com/contest/1609/problem/A*
- *https://codeforces.com/problemset/problem/1427/B*
- *https://codeforces.com/contest/1625/problem/B*
- *https://codeforces.com/contest/1613/problem/B*
- *https://codeforces.com/contest/1418/problem/B* (Reverse)
- *https://codeforces.com/contest/1607/problem/D*
- *https://codeforces.com/problemset/problem/1165/C*

All possible check

It happens that our solution is checking some specific segments.

For example, You are given an array 5 3 4 6 2. Find the minimum number of the given array. If you break the loop when you get 3, is it the correct answer? No. Here we need to traverse the whole array.

So, we need to check every possible solution idea.

Related Problems:

- *https://codeforces.com/gym/101102/problem/F*
- *https://codeforces.com/contest/1194/problem/B*
- *https://codeforces.com/contest/1594/problem/C*
- *https://codeforces.com/contest/1605/problem/C*

Cumulative/Prefix sum / Difference array

Let me give a straight example.

Given an array of size n(n<=100000). There are q(q<=100000) queries. In each query, given l and r, you have to answer the sum of a[l]+a[l+1]+a[l+2]+....+a[r-2]+a[r-1]+a[r].

Sample input:

5
5 7 2 3 10
2
2 5
3 4

Sample output:

22
5

Here are some points of the example,

- If the array size is 100000 and the number of queries is 100000 you cannot traverse from l to r every time. So, we should save the prefix sum of the given array.
- Let's take an array of size n+1 named prefix[n+1].
- Initially prefix[0] = 0
- From the above input the prefix sum is: 5 12 14 17 27
- The answer of the first query and the second query is respectively prefix[5]-prefix[1] and prefix[4]-prefix[2]. **Formula:** prefix[r] - prefix[l-1]

Prefix sum. Resource: https://codeforces.com/blog/entry/88474

Related Problems:

- *https://codeforces.com/contest/1609/problem/C*
- *https://www.codechef.com/JULY21B/problems/MINNOTES*(gcd)

Implementation

The statement helps this time a lot. They have said what to do. Just follow the instructions and solve the problem. To solve this type of problem you need to know STL and Data structures well.

Resource:(Bangla)

L - 01: Course Overview | Standard Template Library (STL) | Competitive Programming

Policy Based Data Structure (PBDS) - Introduction and Basic Usage

Related Problems:

- https://codeforces.com/contest/1609/problem/B
- https://codeforces.com/problemset/problem/1294/C
- https://codeforces.com/contest/1607/problem/A
- https://codeforces.com/contest/1278/problem/A
- https://codeforces.com/contest/1606/problem/B(math)
- https://codeforces.com/contest/1256/problem/D

Two pointers

This technique is very useful. There are a lot of videos on youtube. Watch any videos you want.

Related Problems:

- https://codeforces.com/contest/1582/problem/C

Sorting

Sorting is one of the most used functions in competitive programming. It happens that when we sort the array the problem becomes easier.

Related Problems:

- https://codeforces.com/contest/1607/problem/C

Sieve(prime numbers)

In competitive programming, prime numbers show us a lot of variations.
This Bangla resource will cover a lot of things:

Number Theory #1 (Bangla | বাংলা) Sieve of Eratosthenes | Prime Generator

Related Problems:

- https://atcoder.jp/contests/abc215/tasks/abc215_d
- https://codeforces.com/contest/1586/problem/A

Number of divisors

Related Problems:

- https://toph.co/p/for-loop-ftw

Prime Factorization

Resource: Prime factorization in square root n is not always efficient.
https://www.geeksforgeeks.org/prime-factorization-using-sieve-olog-n-multiple-queries/

Related Problems:

- *https://codeforces.com/contest/1766/problem/D*

GCD

Gcd problems are very interesting. It helps to understand the deep knowledge of math.
Resource: https://forthright48.com/p-cpps-101/

Related Problems:

- *https://codeforces.com/contest/1407/problem/B*
- *https://codeforces.com/contest/1593/problem/D1*

Binary search

Codeforces edu have good problem sets for practice.
Link: https://codeforces.com/edu/course/2/lesson/6

Practice Problem from the video: https://www.youtube.com/watch?v=jp6ZsskrJZI

Related Problems:

- *https://codeforces.com/contest/1436/problem/C*
- *https://codeforces.com/contest/1612/problem/C*
- *https://codeforces.com/contest/1613/problem/C*

Bitmask

Really interesting topic. You just need to know the binary representation of a number including and, or, xor operations.

Resources:
L00 : Course Overview | Bit Manipulation | CodeNCode

Subset sum:

- Explanation
- Implementation

Related Problems:

- *https://atcoder.jp/contests/arc114/tasks/arc114_a*

- *https://codeforces.com/contest/962/problem/C*
- *https://atcoder.jp/contests/abc221/tasks/abc221_c*
- *https://codeforces.com/contest/1594/problem/B*
- *https://codeforces.com/contest/1625/problem/A*
- *https://atcoder.jp/contests/arc129/tasks/arc129_a*
- *https://codeforces.com/problemset/problem/1362/C*

BFS(including 2D BFS), DFS

This topic helps me to understand a lot of things as well as it helps to improve the implementation.

Resources:
L00 : Course Overview | Graph Theory Part 1 | CodeNCode
https://www.shafaetsplanet.com/ (Bangla)

Related Problems:
- *https://atcoder.jp/contests/abc209/tasks/abc209_d*
- *https://codeforces.com/problemset/problem/1037/D*
- *https://codeforces.com/contest/1611/problem/E1*
- *https://codeforces.com/contest/616/problem/C*
- *https://atcoder.jp/contests/abc211/tasks/abc211_d*
- *https://toph.co/p/set-the-meeting*

Thanks for reading this long document. Hope it will help you in some way. Now you should practice more problems that are more than 200 of your current rating. Try to participate in Atcoder contests regularly.
Here I also suggest solving these 100 problems: Medium 100
Some important resources and problem list:

- https://cp-algorithms.com/
- Warm up- Lightoj
- CSES
- FINAL450.xlsx
- Codeforces Edu
- Codeforces Catalog

If you know all of these topics and still struggling I suggest you
- Participate in virtual contests
- Upsolve after every contest and virtual contest
- Try not to watch solutions
- Take help from editorials and comments

Thanks from,
MD. AL-AMIN BHUIYAN
Codeforces Handle: _Alamin_Bhuiyan_
Email: u1804084@student.cuet.ac.bd
Department of CSE, CUET