

## 3-1. Practice Skills – From Rating 1000 to 1400

There are many gray/green coders in Codeforces. They are wanting to be a cyan/blue coder, but actually struggling for it.

Despite this fact, only three things are needed for being rating 1400.

- You can write straight-forward simulation fast. (within 5-10 minutes)
- You can write brute force fast. (within 5-10 minutes)
- You can divide the problem into some cases in your brain or in your paper. (e.g.  $N=2$ ,  $N=3$ , or  $N \geq 4$ )

For example, in Codeforces Round #556, if you can do these three things above, surprisingly you can get even 200<sup>th</sup> place in Div2. This is very exaggerated example, but in Codeforces Round #554 (Div. 2), you can also get 3400<sup>th</sup> place, which participants whose rating less than 1250 increases rating.

On average, if you can do three things above, your rating will reach 1400.

### [[How to practice]]

At first, I suggest you to solve **AtCoder Beginner Contests**.

Though there are many good problems in Codeforces, but if you want to practice to code more simply, you should solve AtCoder.

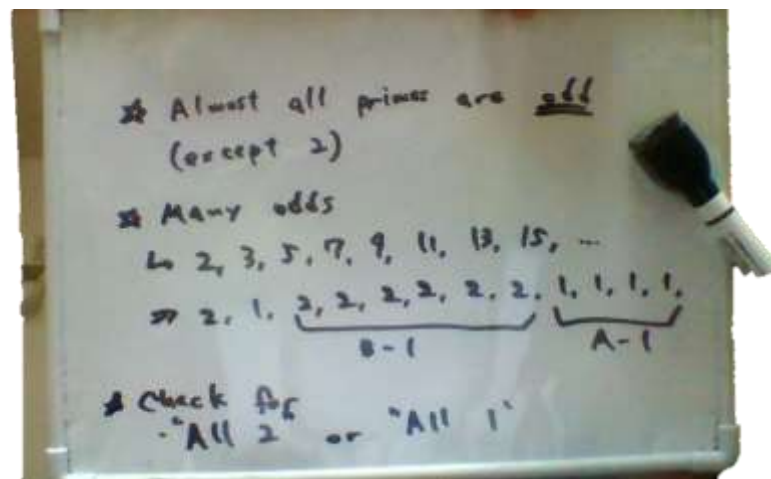
Especially, I recommend to solve problem B or C in AtCoder Beginner Contest. In problem B, you can learn how to write simulations and brute force faster. In problem C, you can learn how to consider the solution and the efficient way to use paper (or notebook) to think about solution.

From AtCoder Beginner Contest 042, English problem statement is added. Since the newest contest is AtCoder Beginner Contest 125, there are 84 Beginner Contests in AtCoder. If you solve all problems of B and C, you would learn much and you will be much stronger.

To solve problems in AtCoder, you can use [AtCoder Problems](#), in which you can know what problem did you solve.

There are some important points when you solve problems in AtCoder.

- If you cannot reach the solution, you should read editorial after 15 minutes' thinking if problem B, and 30 minutes' thinking if problem C. Sadly, in recent ABCs, there could be no English editorial. But you can read sample solution (there are some links of sample code with high probability) to know how to solve this problem.
- Even you can solve a problem, until you get used to code faster, there may be something to learn if you look some source codes written by high-rated coders. So, I recommend to look some simple source codes.
- Especially, when you solve problem C, I recommend you to use paper to consider. For example, in Codeforces Round #556 (Div.2) in C, you can write memo like this:



About this is not paper but whiteboard. Using whiteboard instead of paper is also good.

Since I am a leader of computer club in a high-school, so I have many clubmates, and there are a lot of clubmates which uses this method and improved.