



Competitive Programming

Analyzing and Solving problems

By Gaurav Agarwal



What is it?

- Writing programs to solve problems in a contest environment.
- One has a limited time to solve the problems. Ranging from a few hours to a few days.
- The solution should be efficient in terms of execution time, memory used or in some cases even the length of code.



What's new?

- The code will be executed on an Online Judge (OJ).
- Which means you should write code which conforms to the specific format of input and gives the output in the required format.
- With tough constraints on execution time, it is crucial that you use faster algorithms to avoid TLE(s) on your test cases.
- This requires a good knowledge of time complexities.



For example:

- Say you want to write a function to calculate x^4 .

We can simply solve this with:

```
z=x*x*x*x;
```

```
return z; //required 3 multiplication instructions.
```

but a better solution will be:

```
z=x*x;
```

```
z=z*z;
```

```
return z; //required 2 multiplication instructions.
```



Contd...

- In the previous example, one instruction did not seem to make a difference.
- But if you had to solve the problem for a few million test cases (not uncommon in competitive coding), then you saved yourself from a few million instructions and thereby reduced your execution time correspondingly.
- Using such clever techniques is just a small part of competitive programming 😊
- Thus devising new and better algorithms is crucial for competitive programming.



Understanding the input format

- Input on OJ's is very straightforward. You don't need to prompt for it like we usually do.
- The first line of input is usually the number of test cases, followed by the test cases.



For example:

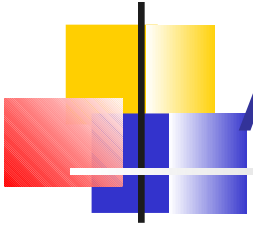
- If the input format states something like: “The first line of input gives the no. of test cases with each test case having 2 numbers n and m.”, then it would look something like this:

3

2 5

6 8

3 9



And its code in C would be:

```
int a,b,c;  
scanf("%d",&a);  
while(a>0)  
{  
    a--;  
    scanf("%d %d",&b,&c);  
}
```

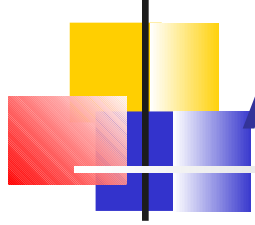



Getting started with competitive coding:

Solving the problems in a coding contest is not easy. Many online judges have problems of similar difficulty and similar environment. A list of such judges are:-

- TopCoder
- SPOJ
- Codechef
- HackerEarth
- HackerRank
- Codeforces

And many more...



A few tips to perform better:

- Have a thorough grasp on the basic concepts first.
- Develop a good understanding of data structures like lists, trees and graphs.
- Try to learn as many new and efficient algorithms and implement them when required.



Some prestigious contests:

- ACM – ICPC
- Google Code Jam
- Topcoder Open
- Facebook Hacker Cup (Recent)

Students who perform well in these contests are some of the finest brains in the world and instantly recognized by top institutions (companies, academia).



What's next?

- Sign up on any of the plethora of online judges and start solving their practice problems.
- You **will** run into a lot of errors in the beginning.
- Also, participate in other online contests arranged by sites like codechef and SPOJ.
- Many companies use these contests to hire students for jobs and internships as well.

Welcome to the world of Competitive programming! 😊