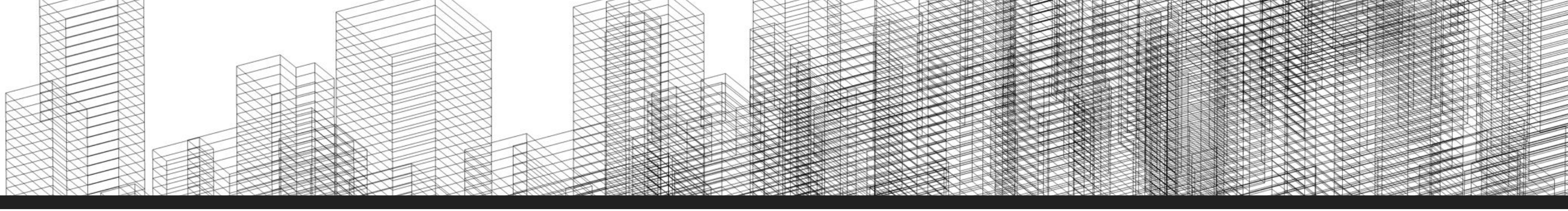
Competitive Coding

Introduction



CP 2019 - Competitive Programming

Instructors

- **Govind P** – thewholeworldfun@gmail.com
- **Kannan K D P** – wizcoder@protonmail.com
- **Ebin J** – ebinpunnethottathil@gmail.com

Goal

- Given a problem, we want to
  - solve it efficiently
  - by using algorithms and data structures,
  - convert our solution into a program,
  - do it as quickly as possible (under pressure)
  - and do it correctly (without bugs)
- This course will exercise this process

How?

- Study common types of problems
- Show common applications of algorithms and data structures you already know from CS205 - Data Structure and CS302 - Design and Analysis of Algorithms
- Introduce other common algorithms and data structures
- Go over some commonly used theory
- Practice problem solving
- Practice programming
- More practice
- More practice

Piazza

- Piazza can be used to ask questions, discuss the problems, create polls.
- https://piazza.com/class/jsx9gimju7n1zs`

Course schedule

Atom #	Topics	Date
1	Arrays and strings	12/03/2019
2	Linked List	19/03/2019
3	Stack Queue	21/03/2019
4	Recursion	TBD
5	DFS,BFS	TBD
6	Hash Tables	TBD
7	Sorting and Searching	TBD
8	Divide and Conquer	TBD
9	Binary tree	TBD

Course schedule

Atom #	Topics	Date
10	Binary search tree	TBD
11	Graph	TBD
12	Backtracking	TBD
13	Dynamic Programming	TBD
14	Greedy Algorithms	TBD
15	Heap	TBD

Problem sets

- Each Atoms covers a topic
- A talk about the topic and problems based on it
- After that you get a set of problems about that topic
- Groups of up to three people can discuss the problems, but each individual must write and hand in their own code.

- Problem sets will be shared in Piazza.
- Each problem set will have 2 to 5 questions.
- Any Q & A can be asked there openly.

Introduction to Competitive Programming

The problems

- Typical programming contest problems
- Usually consists of
    - Problem description
    - Input description
    - Output description
    - Example input/output
    - A time limit in seconds
    - A memory limit in bytes

- You are asked to write a program that solves the problem for all valid inputs
- The program must not exceed time or memory limits

Example problem

Problem description

Write a program that multiplies pairs of integers.

Input description

Input starts with one line containing an integer T, where 1 ≤ T ≤ 100, denoting the number of test cases. Then T lines follow, each containing a test case. Each test case consists of two integers A, B, where −2^20 ≤ A, B ≤ 2^20, separated by a single space.

Output description

For each test case, output one line containing the value of A × B.

Example problem

Sample Input	Sample Output
4	
3 4	12
13 0	0
1 8	8
100 100	10000

Example problem

```
#include <iostream>
    using namespace std;

    int main() {
        int T;
        cin >> T;
        for (int t = 0; t < T; t++) {
            int A, B;
            cin >> A >> B;
            cout << A * B << endl;
        }
        return 0;
}
```

- Is this solution correct?
- What if A = B = 2^20? The output is 0...

No!

Example problem

- When A = B = 2^20, the answer should be 2^40
- Too big to fit in a 32-bit integer, so it overflows
- Using 64-bit integers should be enough

Example problem

```
#include <iostream>
    using namespace std;

    int main() {
        int T;
        cin >> T;
        for (int t = 0; t < T; t++) {
            long long A, B;
            cin >> A >> B;
            cout << A * B << endl;
        }
        return 0;
}
```

- Is this solution correct?

Yes!

Let's Code!!!