Introduction

Atli FF

Árangursrík forritun og lausn verkefna

School of Computer Science

Reykjavík University

Course Overview

Welcome

- T-414-AFLV, Árangursrík forritun og lausn verkefna
- Atli FF, aff6@hi.is, atlif@ru.is
- Arnar Bjarni Arnarson, arnarar@ru.is

Learning goals

- At its heart this course is about problem solving.
- In this course you will learn to take a problem and:
    - analyse the constraints of the problem,
    - take those and find applicable algorithms and data structures,
    - convert those ideas into a functional program,
    - do this quickly and under pressure,
    - producing a program without bugs or other errors.

Getting there

- We will get to this point by going over a number of things:
    - look at common problem types,
    - cover different kinds of problem solving paradigms,
    - show common algorithms and data structures you should know already in action,
    - introduce new less common algorithms and data structures,
    - go over useful theories from a few branches of mathematics,
    - practice solving problems,
    - practice more,
    - practice more,
    - and practice more!

Teaching material

- This course will loosely follow Competitive Programming by Steven Halim
- First edition can be downloaded from the book homepage **https://cpbook.net/**
- A different set of slides for additional reading can also be found at **https://github.com/Kakalinn/tol607g-glaerur**
- Other good material can be found on **cp-algorithms.com**, **codeforces.com**, **wiki.algo.is** and more
- There are plenty of links on Canvas!

Piazza

- Piazza can be used to ask questions
- `https://piazza.com/ru.is/fall2024/t414aflv`
- Before posting questions, read the pinned announcement on what questions can be made publically

Course Schedule

Week no.	Date	Topic
1	19.08	Introduction
2	26.08	Complexity and standard libraries
3	02.09	Ad hoc and complete search
4	09.09	Solving greedily
	14.09	FKHI (10-15 GMT)
5	16.09	Divide and conquer, dynamic programming part 1
6	23.09	Dynamic programming part 2
7	30.09	Data structures
	05.10	NCPC (9-14 GMT)
8	07.10	Graphs part 1
9	14.10	Graphs part 2
10	21.10	Number theory
11	28.10	Combinatorics
12	04.11	Geometry and strings
	22.11	NWERC

Problem Sets

- Each week covers a particular topic
- Groups of up to three people can discuss the problems, but each individual must write and hand in their own code
- We will check for similar submissions, and take action if we think that people are cheating
- Kattis also has a built in anti-cheat feature, which in my personal experience has been plenty good enough to catch a lot of cheaters

Problem Sets cntd.

- Each problem set has ~ 6 problems
- To get a perfect score you need to get at least a certain amount of points
- The problems will give varying number of points depending on difficulty
- The grade is not linear, getting half of the perfect score gets you 7.5
- The deadline for problem sets is always the week's Sunday
- Late handins will not be accepted
- Kattis' verdict is law

Bonus problems

- Each problem set contains 2 challenging bonus problems
- Deadline for all bonus problems is the same as the deadline for the last problem set
- Bonus problems are only included in the final grade if the student passes the course before inclusion
- These can raise the final grade by up to 10%

Other bonuses

- Participating in FKHI and/or NCPC will count as a submitted problem set
- This can effectively replace the lowest or two lowest problem set grades
- Good class/piazza participation can also be rewarded per teacher's discretion

Evaluation

Problem sets	75%
Final exam	25%
Bonus problems/participation	10%
Total	110%

- Remember that bonus problems are only considered if the student passes the course, and is only used to raise the final grade
- A final grade greater than 10 will be reduced down to 10
- The final exam must be passed to pass the course

Problem structure and Kattis

Problem Structure

- A typical programming contest problem usually consists of a
  - 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 $1 \leq T \leq 100$, 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 $-2^{20} \leq A, B \leq 2^{20}$, separated by a single space.

Output description

For each test case, output one line containing the value of $A \times B$.

Example Problem

Sample input	Sample output
4	
3 4	12
13 0	0
1 8	8
100 100	10000

Possible Solution

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

int main() {
    int32_t T;
    cin >> T;
    for(int t = 0; t < T; t++) {
        int32_t A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

Possible Solution

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

int main() {
    int32_t T;
    cin >> T;
    for(int t = 0; t < T; t++) {
        int32_t A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Is this correct?

Possible Solution

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

int main() {
    int32_t T;
    cin >> T;
    for(int t = 0; t < T; t++) {
        int32_t A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Is this correct?
- What if $A = B = 2^{20}$?

Possible Solution

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

int main() {
    int32_t T;
    cin >> T;
    for(int t = 0; t < T; t++) {
        int32_t A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Is this correct?
- What if $A = B = 2^{20}$? The output is 0

Possible Solution

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

int main() {
    int32_t T;
    cin >> T;
    for(int t = 0; t < T; t++) {
        int32_t A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Is this correct? No!
- What if $A = B = 2^{20}$? The output is 0

Fixed Solution

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

int main() {
    int64_t T;
    cin >> T;
    for(int t = 0; t < T; t++) {
        int64_t A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

Fixed Solution

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

int main() {
    int64_t T;
    cin >> T;
    for(int t = 0; t < T; t++) {
        int64_t A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Is this correct?

Fixed Solution

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

int main() {
    int64_t T;
    cin >> T;
    for(int t = 0; t < T; t++) {
        int64_t A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Is this correct?
- The values are at most 2^{20} in absolute value, so their product is at most 2^{40} in absolute value, which fits.

Fixed Solution

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

int main() {
    int64_t T;
    cin >> T;
    for(int t = 0; t < T; t++) {
        int64_t A, B;
        cin >> A >> B;
        cout << A * B << endl;
    }
    return 0;
}
```

- Is this correct? Yes!
- The values are at most 2^{20} in absolute value, so their product is at most 2^{40} in absolute value, which fits.

Automatic Judging

- The problems will be available on Kattis:
- **https://ru.kattis.com/**

- Kattis is an online judge
- You will submit your solutions to Kattis, and get immediate feedback about the solution
- You can submit in any of the supported languages:
    - C, C++, Java, Pythom, C#, Javascript
    - and **many** others

Verdicts

- Feedback is (intentionally) limited
- You will (almost always) receive one of the following verdicts:
    - Accepted (AC)
    - Wrong Answer (WA)
    - Compile Error (CE)
    - Run Time Error (RTE)
    - Time Limit Exceeded (TLE)
    - Memory Limit Exceeded (MLE)
- Neither we nor Kattis will give away info on the test data used to test solutions