

GB21802 - Programming Challenges

Week 0 - Introduction

Claus Aranha
caranha@cs.tsukuba.ac.jp

Department of Computer Science

2019/4/12
(last updated: April 9, 2019)

What is a Programming Challenge? - 0

Consider a string with N characters chosen from G, C, T, A . Ex:

GACACATACAGATTACATTACAGA ... GATACCAGATA

When you receive pair of indexes s and e , calculate the number of "CA"s between N_s and N_e . Ex:

-
-
-

What is a Programming Challenge? - 0

Consider a string with N characters chosen from G, C, T, A . Ex:

GACACATACAGATTACATTACAGA ... GATACCAGATA

When you receive pair of indexes s and e , calculate the number of "CA"s between N_s and N_e . Ex:

- $s = 0, e = 12$ 3 repetitions
-
-

What is a Programming Challenge? - 0

Consider a string with N characters chosen from G, C, T, A . Ex:

GACACATACAGATTACATTACAGA ... GATACCAGATA

When you receive pair of indexes s and e , calculate the number of "CA"s between N_s and N_e . Ex:

- $s = 0, e = 12$ 3 repetitions
- $s = 15, e = 20$ 1 repetition
-

What is a Programming Challenge? - 0

Consider a string with N characters chosen from G, C, T, A . Ex:

GACAC**CATACAG**ATTACATTACAGA ... GATACCAGATA

When you receive pair of indexes s and e , calculate the number of "CA"s between N_s and N_e . Ex:

- $s = 0, e = 12$ 3 repetitions
- $s = 15, e = 20$ 1 repetition
- $s = 4, e = 10$ 2 repetitions

What is a Programming Challenge? - 1

GACACATACAGATTACATTACAGA ... GATACCAGATA

Algorithm Idea 1

Start a counter $c = 0$, loop from N_s to N_e , and every time you find "CA", add to the counter.

Problem: If we have many queries, can we do faster?

What is a Programming Challenge? - 1

GACACATACAGATTACATTACAGA ... GATACCAGATA
00011222233333344444555 ... 88888899999

Algorithm Idea 1

Start a counter $c = 0$, loop from N_s to N_e , and every time you find "CA", add to the counter.

Problem: If we have many queries, can we do faster?

Algorithm Idea 2

Create an auxiliary array A , that keeps the **Cummulative sum** of "CA"s from N_0 to N_i .

If we want to know the answer, we calculate $A_e - A_s$.

What is a Programming Challenge? - 2

- Program Challenges are good for Practicing Algorithms
- Program Challenges are good for Rapid Prototyping
- Program Challenges are used for Work Recruiting
- Program Challenges are also very fun puzzles!

This course is about Programming Challenges

The Goal of this course is **solve programming challenges** to become better at programming.

We will study **algorithms and techniques** that are common in programming competitions

First Things First: Important Notices

Manaba Page

All lecture notes and announcements for this course will be done through MANABA. Access the url below:

https://manaba.tsukuba.ac.jp/ct/course_1149028

Registration Code: 7255921

Language

- Lectures: Japanese
- Slides and materials: English
- Exercises: English
- Questions, Mails and Homework: Any language

About the Lecturer



- **Name:** Claus Aranha;
- **Country:** Brazil;
- **Research:** Artificial Intelligence, Evolutionary Algorithms, Genetic Programming;
- **Language:** Python, R;
- **Hobbies:** Game Programming, Geocaching;
- **webpage:**

<http://conclave.cs.tsukuba.ac.jp>

What is this course about?

You have learned many programming techniques...
...but can you use them?

Course Objective: Learning by Practice

- Every week: Solve 8 programming challenges;
- Choose and implement the **best algorithm** for each problem;
- Be careful with **max time**, and **max memory**;
- We will discuss algorithms, techniques and tricks;

Course Goal:

Improve programming abilities, techniques and familiarity.

Warnings about this class

1- Heavy Workload

- Starts easy, but hard in the end;
- A few hours/week of homework;
- Lots of debugging;
- Hint: Do your homework early!

2- Course Language

- My Japanese is not very good ;-) Let's talk in C++!
- All the course materials are in English;
- You can make your homework in Japanese;
- Practice some English in this course too! :-)

What is a “Programming Challenge”?

A **puzzle** that you solve by **programming**.

Parts of a Programming Challenge:

- Description;
- Standard input;
- Standard output;
- Examples;

Task: Write a program that:

- Reads the input;
- Prints the **correct** output;
- And nothing else!

Tutorial: “Relational Operator” (1)

All challenges are listed at the page:

<http://conclave.cs.tsukuba.ac.jp/lecture/monitor.html>

Let's click on "Relational Operator"

allenges 2016: Problem Monitor

Week 0

Deadline: 88 days, 07:31 hours from now

#	Name	Solved	My Status
1	Relational Operator	0/2	

click to show/hide

Week 1

Deadline: 10 days, 07:31 hours from now

click to show/hide

Tutorial: “Relational Operator” (2)

The link takes you to the UVA (University of Valladolid) homepage (**Please use an ad blocker!**).

Here you can see the problem description, and the links to submit the problem.

The screenshot shows the UVa Online Judge interface. The top navigation bar includes a logo, a search bar, and links for Main Menu, Home Judge, and Browse Problems. The main content area displays the problem details for "11172 - Relational Operator". The problem statement describes the task of checking relationships between two integers. It specifies the input format (an integer t followed by t pairs of integers a and b) and the output format (one line per pair indicating the relationship). The problem page also features a sidebar with links for Quick Submit, Migrate submissions, My Submissions, My Statistics, and a link to the Virtual Contest Service. On the right side, there are buttons for Submit, Statistics, Debug, and PDF.

UVa

Online Judge

Home > Browse Problems

Search

Main Menu

Home

My Account

Contact Us

ACM-ICPC Live Archive

Logout

Online Judge

Quick Submit

Migrate submissions

My Submissions

My Statistics

My uhnrt with Virtual Contest Service

Browse Problems

Quick access, Info and search

Problemssetters' Credits

Root :: Contest Volumes (10000...) :: Volume 112 (11200-11299)

11172 - Relational Operator

Time limit: 3.000 seconds

Some operators checks about the relationship between two values and these operators are called relational operators. Given two numerical values your job is just to find out the relationship between them that is (i) First one is greater than the second (ii) First one is less than the second or (iii) First and second one is equal.

Input

First line of the input file is an integer t ($t < 15$) which denotes how many sets of inputs are there. Each of the next t lines contain two integers a and b ($|a|, |b| < 100000000$).

Output

For each line of input produce one line of output. This line contains any one of the relational operators ' $>$ ', ' $<$ ' or ' $=$ ', which indicates the relation that is appropriate for the given two numbers.

Sample Input

3
10 20

Submit Statistics
Debug PDF

Tutorial: “Relational Operator” (3)

Description

*Some operators checks about the relationship between two values and these operators are called relational operators. Given two numerical values **your job is** just to find out the relationship between them that is (i) First one is greater than the second (ii) First one is less than the second or (iii) First and second one is equal.*

- Reading the description can be the hardest part!
- For this challenge, you just need to:
 - If the first number is bigger than the second; print ">"
 - If the first number is smaller than the second; print "<"
 - If both numbers are equal, print "="

Very easy!

Tutorial: “Relational Operator” (4)

Input and Output

Input

First line of the input file is an integer t ($t < 15$) which denotes how many sets of inputs are there. Each of the next t lines contain two integers a and b ($|a|, |b| < 1000000001$).

Output

For each line of input produce one line of output. This line contains any one of the relational operators " $>$ ", " $<$ " or " $=$ ", which indicates the relation that is appropriate for the given two numbers.

- It is important to read the **size** of the problem!
- Pay attention to the **shape** of the output!

Tutorial: “Relational Operator” (5)

Examples

Sample Input

```
3
10 20
20 10
10 10
```

Sample Output

```
<
>
=
```

- Use the samples to test/debug your program.
- Use your own samples too!
- Use the samples to understand the challenge.

Solution: C++

```
// UVA 11172 - Relational Operator
// Test if a is bigger, smaller or equal to b

#include <iostream>
using namespace std;

int main()
{
    int n; long a, b;

    cin >> n;
    for (; n > 0; n--)
    {
        cin >> a >> b;
        if (a > b) cout << ">\n";
        if (a < b) cout << "<\n";
        if (a == b) cout << "=\\n";
    }
}
```

Solution: Python

```
n = int(input())

while (n > 0):
    line = input()
    tokens = line.split()
    a,b = int(tokens[0]),int(tokens[1])

    if a > b: print(">")
    if a < b: print("<")
    if a == b: print("==")

    n -= 1
```

Solution: Java

```
import java.io.*;
class Main
{
    public static void main(String args[])
    {
        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter stdout = new BufferedWriter(new OutputStreamWriter(System.out));
        try {
            String line;
            line = stdin.readLine();
            int n = Integer.parseInt(line);

            for (int i = 0; i < n; i++)
            {
                line = stdin.readLine();
                String[] tokens = line.split("\s+");
                long a = Integer.parseInt(tokens[0]);
                long b = Integer.parseInt(tokens[1]);

                if (a > b)
                    stdout.write(">\n");
                if (a < b)
                    stdout.write("<\n");
                if (a == b)
                    stdout.write("=\n");
                stdout.flush();
            }
            stdout.close();
        } catch (IOException ioe) { System.out.println("I/O Exception"); }
    }
}
```

Java Solution – Keep in Mind

- All code must be in the one file;
- The `static main` method must be in `Main` class.
- Do not use public classes. Even Main must be non public.
- Use Buffered I/O for faster input/output.

Submitting the problem to UVA

11172 - Relational Operator

Language

- ANSI C 5.3.0 - GNU C Compiler with options: -lm -lcrypt -O2 -pipe -ansi -DONLINE_JUDGE
- JAVA 1.8.0 - OpenJDK Java
- C++ 5.3.0 - GNU C++ Compiler with options: -lm -lcrypt -O2 -pipe -DONLINE_JUDGE
- PASCAL 3.0.0 - Free Pascal Compiler
- C++11 5.3.0 - GNU C++ Compiler with options: -lm -lcrypt -O2 -std=c++11 -pipe -DONLINE_JUDGE
- PYTH3 3.5.1 - Python 3

Paste your code...

...or upload it

No file chosen

- After you complete the program, use the **submit** button in the UVA page;
- Choose your language and submit your code;
- You can choose C, C++, Java or Python.

Submitting the problem to UVA

My Submissions

#	Problem	Verdict	Language	Run Time	Submission Date
17182419	1124 Celebrity jeopardy	Accepted	C++	0.000	2016-04-11 06:42:30
17181459	10141 Request for Proposal	Compilation error	C++11	0.000	2016-04-11 01:36:46
17181444	11498 Division of Nlogonia	Accepted	C++11	0.000	2016-04-11 01:30:43
17071417	102 Ecological Bin Packing	Compilation error	C++11	0.000	2016-03-23 09:21:55
17070667	161 Traffic Lights	Accepted	C++	0.000	2016-03-23 07:24:56
16607686	489 Hangman Judge	Accepted	C++	0.349	2015-12-20 03:52:45
16607670	489 Hangman Judge	Wrong answer	C++	0.335	2015-12-20 03:47:01
16607649	489 Hangman Judge	Runtime error	C++	0.000	2015-12-20 03:40:51

- UVA has an [Automated Judge](#).
- After 2 5 minutes, you should receive an e-mail with the result.
- All your results can be seen at "My Submissions" Page.

Submitting the problem to UVA

These are the possible results:

- Accepted: Your program is correct! Congratulations!
- Presentation Error: Small mistake in number of spaces. Congratulations!
- Wrong Answer: Your program is incorrect. Time to debug.
- Time Limit Exceeded: Your program is too slow.
- Memory limit exceeded: Your program uses too much memory.
- Runtime Error: Your program crashed (segmentation fault!)

Problem Monitor Page

Challenges 2016: Problem Monitor

Week 0

Deadline: 11 days, 00:18 hours from now

#	Name	Solved	My Status
1	Division of Nlogonia	1/2	Accepted
2	Cancer or Scorpio	0/2	Not submitted
3	The 3n + 1 problem	0/2	Not submitted
4	Request for Proposal	0/2	Not accepted

click to show/hide

- All Problems and results;
- All Deadlines;

Submitting the problem to MANABA

After you finish the problems listed in the monitor, you need to submit your source code and a comment file as a zip package to MANABA.

s2015XXXXXX-weekYY.zip

- problem1.cpp
- problem2.cpp
- problem5.cpp
- kaisetsu.txt

Attention

Submission to the UVA judge without a submission to MANABA will not be accepted!

Course Rules

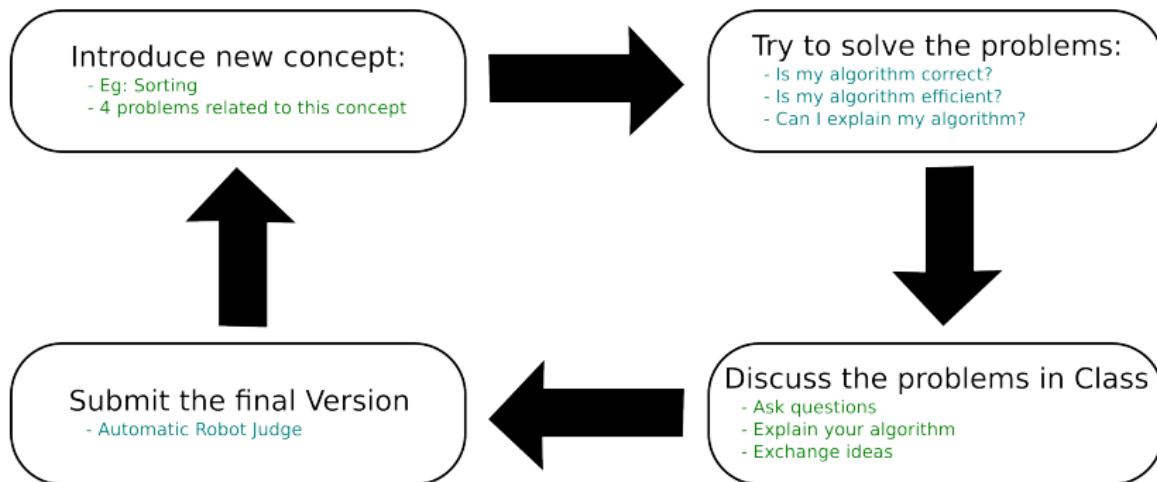
Two classes per week

- Each week has a theme (Graphs, Maths, etc...)
- Friday Class: Introduction
- Monday Class: Problem Solving and Q&A

Solving Problems

- Every week there are 8 programming assignments;
- Deadline is Thursday 23:59
 - UVA Submission
 - MANABA Submission

Outline



Grading Algorithm

Your Grade: **Base Grade** +**Bonus** -**Penalty**

- **Base Grade:**
 - You solved 2/8 problem/week: "C"
 - You solved 3/8 problems/week: "B"
 - You solved 4/8 problems/week: "A"
- **+Bonus**
 - 5% of best student in each category.
 - Special collaboration to the class.
- **-Penalty**
 - More than 25% of homework submitted late.

Evaluating and Grading: KAISETSU file

- Submit a TEXT (not word) file with your impressions of the problems, class, life, together with your code.
- Kaisetsu file can be in any Japanese or English.

Example

Name: Claus, ID: 98884735

Problem 1:

This problem was very easy, but I did a stupid mistake and it took me three hours to solve. I found the solution when I tried a new test case.

Problem 2:

This problem was very hard, and I was very hungry, so I gave up.

Evaluation and Grading (5) – about plagiarism

The assignments are **individual**. Use your **own** strength to solve the programs.

GOOD

- Ask for ideas to your friends;
- Ask for ideas in the MANABA forum;
- Ask for help with a bug;

BAD

- Copy a solution from the internet;
- Copy a solution from your friends;
- Give your code to a friend;

Plagiarism will result in course failure, and possibly worse.

Useful Links

- [Manaba Page](#): All the class material will be here. Access Code is: 7255921
- [UVA Online Judge](#): Use this page to submit your problems.
Make an account and list the username on MANABA
- [Problem Monitor](#): Use this page to check deadlines and weekly problems.
- [Github Repository](#): Working directory for lecture notes.
Send me PR, issues!
- [uDebug](#): Web service that generates test inputs and test outputs for UVA problems. Useful tool for this course.

Course Book

- Competitive Programming, 3rd Edition (<http://cpbook.net>)
- For suggestions of books in Japanese, please check the Manaba materials!

uDebug Tool

uDebug generates outputs for many different debugs. It can help you check why your program is wrong.

<https://www.udbug.com/>



The uDebug logo features the word "uDebug" in a bold, black, sans-serif font. A small, stylized icon resembling a red bug or a drop with a grid pattern is positioned to the right of the letter "g".

Search for a problem you've solved, provide input, and get accepted output!

[8299 problems and counting!](#)

Contact the professor

- e-mail: caranha@cs.tsukuba.ac.jp
- website: <http://conclave.cs.tsukuba.ac.jp>
- Room: SB1012 – Send an e-mail and we can talk!

Both English and Japanese are okay!

What to do this weekend?

- Create an account on UVA (if you already have an account, you can use that)
- Submit your account name to the MANABA
- Ask any other questions you want to know!

Participate in ICPC!

- Fun challenges to choose the world champion!
- Teams of 3 people!
- Registration Deadline Early of June!

