

# GB21802 - Programming Challenges

## Week 0 - Introduction

Claus Aranha  
caranha@cs.tsukuba.ac.jp

Department of Computer Science

March 27, 2014

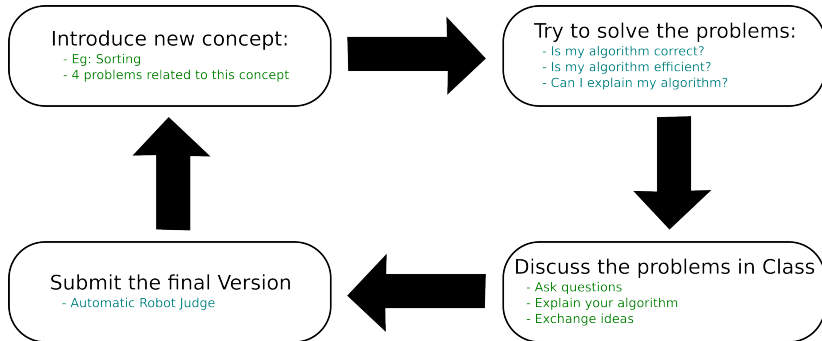
# What is this course about?

## A “Strange” Course

**Goal:** Improve the understanding of algorithms and programming techniques.

**Method:** Solve short and hard problems using well known algorithms

# Course Outline



# Course Languages

- Program Language: C, C++, Java, Pascal
- Spoken Language: Japanese
- Materials and Problems: English
- Reports and Questions: Both!

# What are programming challenges?

Short (but sometimes hard) problem involving algorithms

## Components

- **Problem Outline**
- Example Data
- Example Result
- Hidden Data
- Judge Result

Start with an integer  $n$ . If  $n$  is even, divide by 2. If  $n$  is odd, multiply by 3 and add 1.

Repeat this process with the new value of  $n$ , terminating when  $n = 1$ . For example:

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

In the example above, the cycle length of 22 is 16. Given any two numbers  $i$  and  $j$ , you are to determine the maximum cycle length over all numbers between  $i$  and  $j$ , including both endpoints.

# What are programming challenges?

Short (but sometimes hard) problem involving algorithms

## Components

- Problem Outline
- Example Data
- Example Result
- Hidden Data
- Judge Result

### Input

The input will consist of a series of pairs of integers  $i$  and  $j$ , one pair of integers per line. All integers will be less than 1,000,000 and greater than 0.

### Output

For each pair of input integers  $i$  and  $j$ , output  $i$ ,  $j$  in the same order in which they appeared in the input and then the maximum cycle length for integers between and including  $i$  and  $j$ .

# What are programming challenges?

Short (but sometimes hard) problem involving algorithms

## Components

- Problem Outline
- Example Data
- Example Result
- Hidden Data
- Judge Result

### Sample Input

```
1 10
100 200
201 210
900 1000
```

### Sample Output

```
1 10 20
100 200 125
201 210 89
900 1000 174
```

# What are programming challenges?

Short (but sometimes hard) problem involving algorithms

## Components

- Problem Outline
- Example Data
- Example Result
- **Hidden Data**
- Judge Result

### Sample Input

```
1 10
100 200
201 210
900 1000
```

### Sample Output

```
1 10 20
100 200 125
201 210 89
900 1000 174
```



# What are programming challenges?

Short (but sometimes hard) problem involving algorithms

## Components

- Problem Outline
- Example Data
- Example Result
- Hidden Data
- **Judge Result**

Accepted

Rejected

Time Limited Exceeded (TLE)

# How the Classes will work

## Monday

**Problem presentation:** The week theme will be presented, and 4 problems regarding that theme will be shown.

## Friday

**Problem discussion:** Students discuss together questions about problems and how to solve them.

## Deadline

Deadline for program submission is **Sunday, Midnight**  
Programs submitted after the deadline are accepted with penalty.

# Evaluation and Grading

Evaluation is based on solving the programs, and participation in class.

- C: One problem per class;
- B: Two problems per class, or 20 problems;
- A: Three problems per class, or 30 problems;

## Bonus: Grade Up

Good participation in class and good Comments in code.

## Penalty: Grade Down

More than 25% late problems.

# How to submit problems - 1

## Problem Submission System

- 1 Make an account at  
`http://www.programming-challenges.com;`  
(If possible use your Student Number as ID)
- 2 Send your ID to the professor by e-mail;  
`mailto:caranha@cs.tsukuba.ac.jp`
- 3 You will be added to the classroom  
[Tsukuba Programming Challenges 2015;](#)

# How to submit problems - 2

## Problem Submission System

- ④ Click “Joined Classrooms”, select [Tsukuba Programming Challenges 2015](#);
- ⑤ Click the name of the problem for a description, then “Submit” to send your code.
- ⑥ Choose the language; upload a file or paste your code.
- ⑦ Wait for the response from the Judge!

# Some notes about program submission

Please Keep in Mind:

- Don't copy programs from the internet, or from your friends;
- It is okay to copy ideas from the internet or your friends;  
If you do, mention it in the comments
- Add some commentary on top of the program, explaining what you did, what went right, or what went wrong.

# Some notes about program submission

## Good Comment

```
/**  
 * I used quicksort to solve this problem.  
 * I sorted the age of the persons in the input.  
 * To make it faster, people with the same age were  
 * removed from the data.  
 */
```

## Bad Comment

```
/**  
 * Quicksort.  
 */
```

# How the Judge Works

Accepted

Congratulations!

Wrong Answer

Your answer does not match with the judge's answer.  
Remember to check for worst-case scenarios!

Time Limited Exceeded

Your algorithm is too slow. Think about computational efficiency.

Compilation Error, Runtime Error, etc.



# OMAKE I: Programming Contests

- What are Programming Contests?
- Examples: ACM-ICPC, TOPCODER, ATCODER, ...

## OMAKE II: Next week's problems

- $3n+1$  Problem
- Minesweeper
- The Trip
- Interpreter

A proper introduction to these problems will be made next week!