

Algorithms for Programming Contests - Week -1

Prof. Dr. Javier Esparza
Mikhail Raskin, Tobias Meggendorfer,
Christoph Welzel, A. R. Balasubramanian,
`conpra@in.tum.de`

22. Januar 2020

General Concept

- Theoretical background about several concepts and algorithms.
- Implementing algorithms is a useful skill, for CS research, other research, software development. . . sometimes for hobbies.
- Contest-format algorithm implementation practical course.

The Problem Sets

- Available on the course homepage, accessible with password only (you can also see the current tasks in the contest system — zip file of sample inputs only available on the course page).
- Usually, five problems per week:
 - two easy (4 Points each),
 - two medium (6 Points each),
 - one hard (8 Points).
- Each week's problems are (mostly) about the topics, algorithms and concepts from that week's lecture.
- There will be hints in the lectures.

Have a Problem with a Problem?

Should difficulties arise: Ask questions! Come to our offices or write a clarification request!

Grading

Your final grade will be determined by how many points you earned, as well as an oral exam at the end of the semester. The oral exam will account for 40% of the grade.

The (tentative) key for grading the problems is the following:

Percentage	Grade
$\geq 90\%$	1.0
$\geq 85\%$	1.3
$\geq 80\%$	1.7
$\geq 75\%$	2.0
$\geq 70\%$	2.3
$\geq 65\%$	2.7
$\geq 60\%$	3.0
$\geq 55\%$	3.3
$\geq 50\%$	3.7
$\geq 40\%$	4.0

Topics (preliminary list)

- 1 Introduction
- 2 Data Structures (UF, Binary Search, Graphs)
- 3 Graphs, Minimum Spanning Trees, DFS, BFS
- 4 Shortest Paths
- 5 Maximum Flows
- 6 Brute Force / Backtracking
- 7 Greedy
- 8 Dynamic Programming
- 9 Geometry
- 10 Number Theory
- 11 Trees
- 12 Contest
- 13 Conclusion

Official DOMjudge System

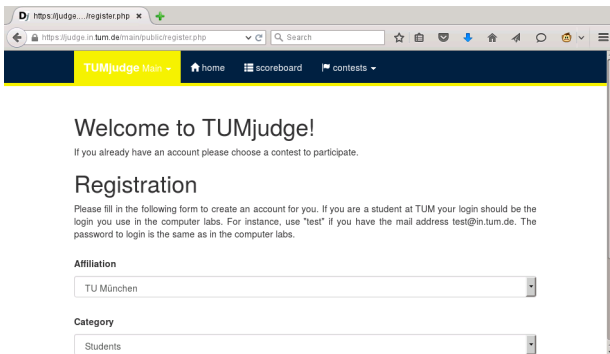
- In use for programming contests such as the GCPC or the ICPC.
- On the web:

TUMjudge

<https://judge.in.tum.de/conpra/>

Registration

- Registration necessary.



The screenshot shows a web browser window with the URL `https://judge.in.tum.de/main/public/register.php`. The browser's address bar also shows `https://judge.../register.php`. The page has a dark blue header with a yellow bar containing the text "TUMjudge Main". Below the header, there are navigation links: "home", "scoreboard", and "contests". The main content area has a heading "Welcome to TUMjudge!" followed by the text "If you already have an account please choose a contest to participate." Below this is a heading "Registration" and a paragraph explaining the registration process: "Please fill in the following form to create an account for you. If you are a student at TUM your login should be the login you use in the computer labs. For instance, use 'test' if you have the mail address test@in.tum.de. The password to login is the same as in the computer labs." The form consists of two dropdown menus. The first is labeled "Affiliation" and has "TU München" selected. The second is labeled "Category" and has "Students" selected.

Affiliation

TU München

Category

Students

Problem structure

A problem consists of several parts:

- name, abbreviation, difficulty,
- problem author,
- problem statement,
- input format specification,
- output format specification,
- constraints,
- sample input and output.

SS15N01A Hello World!

Author: Stefan Toman

This is probably the first problem you will solve and it should help you set up and test your system. Solve this problem first to make sure everything is in place.

We would like to introduce you to Lea. You will meet her in many of the problems you will solve. After reading all of them you will know her quite well.

Lea is a very friendly person who likes to say hello to everybody, but she doesn't want to say the same thing to every person she meets. Therefore, she never knows what to say. For greeting Bob it is appropriate to say "Hello Bob!", whereas for greeting Peter it is better to say "Hello Peter!". Help her and tell her which sentence to use.

Input

The first line of the input contains an integer t , t test cases follow.

Each test case consists of a single line containing a name *name*.

Output

For each test case, print a line containing "Case i : Hello *name*!" where i is its number, starting at 1. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$.
- No name will contain whitespaces.
- The names' lengths will be at most 100.

Sample Data

Input

```
1 2
2 Bob
3 Peter
```

Output

```
1 Case #1: Hello Bob!
2 Case #2: Hello Peter!
```


Submitting programs

Submitting program is done on the TUMjudge web interface entirely.

- No files to be sent via e-mail etc.
- Only source code files are uploaded, no .class files or similar.

Submit

- Choose files to be uploaded by Drag-and-Drop or in the menu “Choose Files”,
- Choose problem,
- Choose language (unless already chosen automatically),
- “submit”,
- F5, F5, F5, F5, ... (the page actually reloads automatically)

Judging

The TUMjudge

- compiles,
- executes,
- tests

the submission against several test cases. As long as the TUMjudge is working on a submission, the submission's status is "PENDING".

The submission is treated instantaneously and the TUMjudge (usually) announces its verdict within a few moments.

Judging

The following verdicts can occur:

CORRECT

The submission successfully solved all the test cases.

COMPILER-ERROR

The submission could not be compiled. The exact error message can be seen on the submission's detail page.

NO-OUTPUT

The submission does not produce any output. Be sure to output to "standard out".

Judging

TIMELIMIT

The submission runs longer than the maximal allowed time and was terminated.

Possible reasons:

- The submission runs in an endless loop.
- The submission is not efficient enough.

RUN-ERROR

An error occurred during the submission's execution.

Possible reasons:

- Division by 0.
- Incorrectly addressing memory locations, e.g. `ArrayIndexOutOfBoundsException`.
- Using more memory than the allowed memory limit.

Judging

WRONG-ANSWER

The submission's output is incorrect.

Possible reasons:

- The answer is just wrong.
- The answer does not conform to the output format specification given on the problem set.
- The answer is not exact enough (e.g. with floating point answers with a desired precision).

TOO-LATE

The program was submitted after the submission deadline. It is stored in the system, but no longer processed.

Scoreboard

Scoreboard Algorithms for Programming Contests SS2015 - Week 10

final standings

RANK	TEAM	SCORE	A	B	C	D	E
1	Jens Woelfle TU Mönchen (3 Wkt)	28 3405	1/68	1/111	1/126	3/362	3/421
2	Thomas Tangl TU Mönchen (5 Wkt)	28 5895	1/0	5/0	1/0	3/1114	1/1181
3	Markus Hasenöhrl TU Mönchen (3 Wkt)	28 7251	1/0	1/0	1/0	9/1242	1/1209
4	Karl Kraus TU Mönchen	28 9461	3/0	5/5892	1/5	1/82	1/183
5	Michael Schreier TU Mönchen	28 13449	1/0	1/2	13/55	9/1106	1/277
6	Rustem Bekmukhametov TU Mönchen	28 15914	3/144	7/229	1/257	3/6218	5/666
7	Philipp Becker Ehmck TU Mönchen	28 19329	1/206	1/248	1/269	7/8997	1/609
8	Christian Buttner TU Mönchen	28 20531	1/177	5/5580	1/1237	1/5567	1/5570
9	Hamidreza Rizeh TU Mönchen	28 25240	1/886	1/987	1/988	25/5582	3/997
10	Rob Coekaerts TU Mönchen	28 28218	1/5564	1/5565	1/5585	1/5715	1/5598
11	Anjum Parvez Ali TU Mönchen	28 29878	5/443	2/5886	1/5855	5/6328	1/5895

ws14 Highlight All Match Case 1 of 13 matches

Different background colors indicate different outcomes:



Problem solved.



Problem solved first.



Incorrect submission(s).



Submission in pending status.



No submissions.

Scoreboard

Order (tie-breakers):

- ① number of solved problems,
- ② score:
 - per problem: (number of incorrect submissions) * (penalty time) + (time for the first correct submission),
 - penalty time = 600, i.e., 10 hours,
 - e.g. **3/1820** indicates: the problem was solved with 3 submissions, with a total penalty time of 1820.

You can submit any number of times to solve a problem!

Each week's score itself (apart from the number of problem solved) does not change the grading at the end of the semester, it only affects the position in the scoreboard of that week.

Scoreboard

Anybody who does not want to be seen in the public scoreboard, must choose the invisibility option during the registration (in “Category”).

Clarifications

- Messages to the system administrators, i.e., the teaching assistants and/or tutors.
- Sent via the “request clarification” form on the overview page.
- Used for questions about the problems or about the system in general.
- Please choose a subject accordingly: either “general” or the specific problem.
- Depending on the actual question, the answer is only visible to the persons who sent the question, or it is published to all users of the system.
- The answer (along with the question) can be seen on the right side on the overview page.

Clarifications

Stefan Toman

https://judge.in.tum.de/conprateam/

TUMjudge ConPra

home course scoreboard news help contests toman

Algorithms for Programming Contests SS2015 - Week 10

final standings

RANK	TEAM	SCORE	A	B	C	D	E
1	Stefan Toman TU München	5 8632	1/0	1/0	1/0	1/8632	1/0

Submissions

Browse... No files selected. problem

language submit cancel

time	problem	lang	result
09.07.2015 11:43	E	JAVA	TOO-LATE
08.07.2015 14:18	B	CPP	CORRECT
08.07.2015 11:52	D	CPP	CORRECT
01.07.2015 11:24	E	JAVA	CORRECT

Clarifications

time	from	to	subject	text
09.07.2015 12:12	Jury	All	problem E	Dear students, here is the link to Karl's website for drawing the fractals. ...
02.07.2015 12:41	Jury	All	problem E	Dear students, please remember that we are looking forward to get your great ...

Clarification Requests

No clarification requests.

request clarification

TUMjudge version 5.0.0.0, a fork of DOMjudge version 5.0.1 Imprint / Changelog

Discussion forum

Sometimes you want to discuss with other students instead of sending clarification requests.

Please discuss problem statements, corner cases, algorithms and approaches. The code should be your own.

You are welcome to use the forum instance with a subsection dedicated to the course: <https://judge.in.tum.de/discuss/c/conpra>

You are welcome to discuss and post test cases if you want to
(But please do not post and do not ask for code snippets)

Understanding Problems

- Read the problem statement very carefully.
- Also the constraints, think about special cases:
 - E.g. if there are negative values or 0 allowed, then there is probably a test case for that.
 - E.g. special characters or a space when dealing with strings.
 - ...

Solving Problems

- Code efficiently.
 - Think about which data types to use.
 - Sometimes arrays might not have to be two- or three-dimensional.
 - Implement algorithms given in the lecture with their amortized running times.
- Look carefully at the input and output specifications and let your program be conform to those!
- Remove all debug messages before submitting.
- Write comments!

Code from the Internet

- You are allowed to download and include a library, but you need to cite the correct source.
- Do this by putting a comment in your code stating the url or similar.
- Including other people's published code without a reference, or using other students' code, is a rule violation
- However, we advise you to code on your own as it improves the understanding about the algorithms involved. You probably need this in subsequent problems anyway.

Half Points

If the judge does not accept your solution but you are sure you solved it correctly, use the “request clarification” option. In your request include:

- the name of the problem (by selecting it in the subject field),
- a verbose description of your approach to solve the problem,
- the time you submitted the solution we should judge.

We will check your submission and award you half the points if there is only a minor flaw in your code.

We might use submissions from half-point requests (anonymised) as realistic examples of subtle problems in the lectures and debugging demo sessions.

Languages

- C++, Java — supported since forever
- Python3 — PyPy3, apparently sufficient for all problems Might require more careful implementation than C++/Java
- Julia — seems to work, and experiment

Registration

We want to select the students who will complete the course. . .
Please show you can do simple problems
algorithms. . . we will talk about them

Equal annoyance for local and exchange students:
SPOJ as a neutral platform
No need to give real name to SPOJ

Procedure on course web page:
<https://www7.in.tum.de/um/courses/praktika/conpra/SS20/>