Programming Competitions for ITESM: Introduction

Maxim Buzdalov, Niyaz Nigmatullin

September 7, 2015

Outline

Course website

http://neerc.ifmo.ru/trains/itesm

Programming competitions

Programming competitions

What are they about?

Programming competitions

What are they about?

- Several problems are given

Programming competitions

What are they about?

- Several problems are given
  - input data format

Programming competitions

What are they about?

- Several problems are given
  - input data format
  - output data format

Programming competitions

What are they about?

- Several problems are given
  - input data format
  - output data format
  - what to do with that

Programming competitions

What are they about?

- Several problems are given
  - input data format
  - output data format
  - what to do with that
- Solve as many problems as you can

Programming competitions

What are they about?

- Several problems are given
  - input data format
  - output data format
  - what to do with that
- Solve as many problems as you can
  - Write a program

Programming competitions

What are they about?

- Several problems are given
  - input data format
  - output data format
  - what to do with that
- Solve as many problems as you can
  - Write a program
  - Efficiently: fit time and memory limits

Programming competitions

What are they about?

- Several problems are given
  - input data format
  - output data format
  - what to do with that
- Solve as many problems as you can
  - Write a program
  - Efficiently: fit time and memory limits
  - Program is tested on several test cases

Programming competitions

What are they about?

- Several problems are given
  - input data format
  - output data format
  - what to do with that
- Solve as many problems as you can
  - Write a program
  - Efficiently: fit time and memory limits
  - Program is tested on several test cases
    - Compilation error

Programming competitions

What are they about?

- Several problems are given
  - input data format
  - output data format
  - what to do with that
- Solve as many problems as you can
  - Write a program
  - Efficiently: fit time and memory limits
  - Program is tested on several test cases
    - Compilation error
    - Wrong answer, test XX

Programming competitions

What are they about?

- Several problems are given
  - input data format
  - output data format
  - what to do with that
- Solve as many problems as you can
  - Write a program
  - Efficiently: fit time and memory limits
  - Program is tested on several test cases
    - Compilation error
    - Wrong answer, test XX
    - Time limit exceeded, test XX

Programming competitions

What are they about?

- Several problems are given
  - input data format
  - output data format
  - what to do with that
- Solve as many problems as you can
  - Write a program
  - Efficiently: fit time and memory limits
  - Program is tested on several test cases
    - Compilation error
    - Wrong answer, test XX
    - Time limit exceeded, test XX
    - Security violation, test XX

Programming competitions

What are they about?

- Several problems are given
  - input data format
  - output data format
  - what to do with that
- Solve as many problems as you can
  - Write a program
  - Efficiently: fit time and memory limits
  - Program is tested on several test cases
    - Compilation error
    - Wrong answer, test XX
    - Time limit exceeded, test XX
    - Security violation, test XX
    - Accepted

This course

This course

The aim

This course

The aim

- Learn how to solve problems efficiently

This course

The aim

- Learn how to solve problems efficiently
  - invent an efficient algorithm

This course

The aim

- Learn how to solve problems efficiently
  - invent an efficient algorithm
  - write an efficient program

This course

The aim

- Learn how to solve problems efficiently
  - invent an efficient algorithm
  - write an efficient program
  - use human/computer resources efficiently

This course

The aim

- Learn how to solve problems efficiently
  - invent an efficient algorithm
  - write an efficient program
  - use human/computer resources efficiently
- Learn how to implement solutions correctly

This course

The aim

- Learn how to solve problems efficiently
  - invent an efficient algorithm
  - write an efficient program
  - use human/computer resources efficiently
- Learn how to implement solutions correctly
- Learn how to debug your implementations

This course

The aim

- Learn how to solve problems efficiently
  - invent an efficient algorithm
  - write an efficient program
  - use human/computer resources efficiently
- Learn how to implement solutions correctly
- Learn how to debug your implementations
- Learn to read code from the paper

This course

This course

The structure

This course

The structure

- Contests

This course

The structure

- Contests
  - thematic problems (graphs, CG, DP)

This course

The structure

- Contests
  - thematic problems (graphs, CG, DP)
  - real problems (high school competitions)

This course

The structure

- Contests
  - thematic problems (graphs, CG, DP)
  - real problems (high school competitions)
- Problem analysis

This course

The structure

- Contests
  - thematic problems (graphs, CG, DP)
  - real problems (high school competitions)
- Problem analysis
- Lectures

This course

The structure

- Contests
  - thematic problems (graphs, CG, DP)
  - real problems (high school competitions)
- Problem analysis
- Lectures
  - coding tips and tricks, team strategies, ...

Programming environment

Programming environment

Available languages

Programming environment

Available languages

- C, C++, Java (as at World Finals)

Common things for problems

Common things for problems

- Testing machines: AMD Phenom II X4 955, 3.2GHz, 8G RAM

Common things for problems

- Testing machines: AMD Phenom II X4 955, 3.2GHz, 8G RAM
- Use files or stdin/stdout for input/output

Common things for problems

- Testing machines: AMD Phenom II X4 955, 3.2GHz, 8G RAM
- Use files or stdin/stdout for input/output
  - typically `something.in`/`something.out`, if using files

Common things for problems

- Testing machines: AMD Phenom II X4 955, 3.2GHz, 8G RAM
- Use files or stdin/stdout for input/output
  - typically `something.in`/`something.out`, if using files
  - "security violation" if waiting for `stdin`, if using files

Common things for problems

- Testing machines: AMD Phenom II X4 955, 3.2GHz, 8G RAM
- Use files or stdin/stdout for input/output
  - typically `something.in`/`something.out`, if using files
  - "security violation" if waiting for `stdin`, if using files
  - be careful when reading problem statement

Common things for problems

- Testing machines: AMD Phenom II X4 955, 3.2GHz, 8G RAM
- Use files or stdin/stdout for input/output
  - typically `something.in`/`something.out`, if using files
  - "security violation" if waiting for `stdin`, if using files
  - be careful when reading problem statement
- Time limit: see problem statement

Common things for problems

- Testing machines: AMD Phenom II X4 955, 3.2GHz, 8G RAM
- Use files or stdin/stdout for input/output
  - typically `something.in`/`something.out`, if using files
  - "security violation" if waiting for `stdin`, if using files
  - be careful when reading problem statement
- Time limit: see problem statement
- Memory limit: typically 256 megabytes

Common things for problems

- Testing machines: AMD Phenom II X4 955, 3.2GHz, 8G RAM
- Use files or stdin/stdout for input/output
  - typically `something.in`/`something.out`, if using files
  - "security violation" if waiting for `stdin`, if using files
  - be careful when reading problem statement
- Time limit: see problem statement
- Memory limit: typically 256 megabytes
- We report test numbers

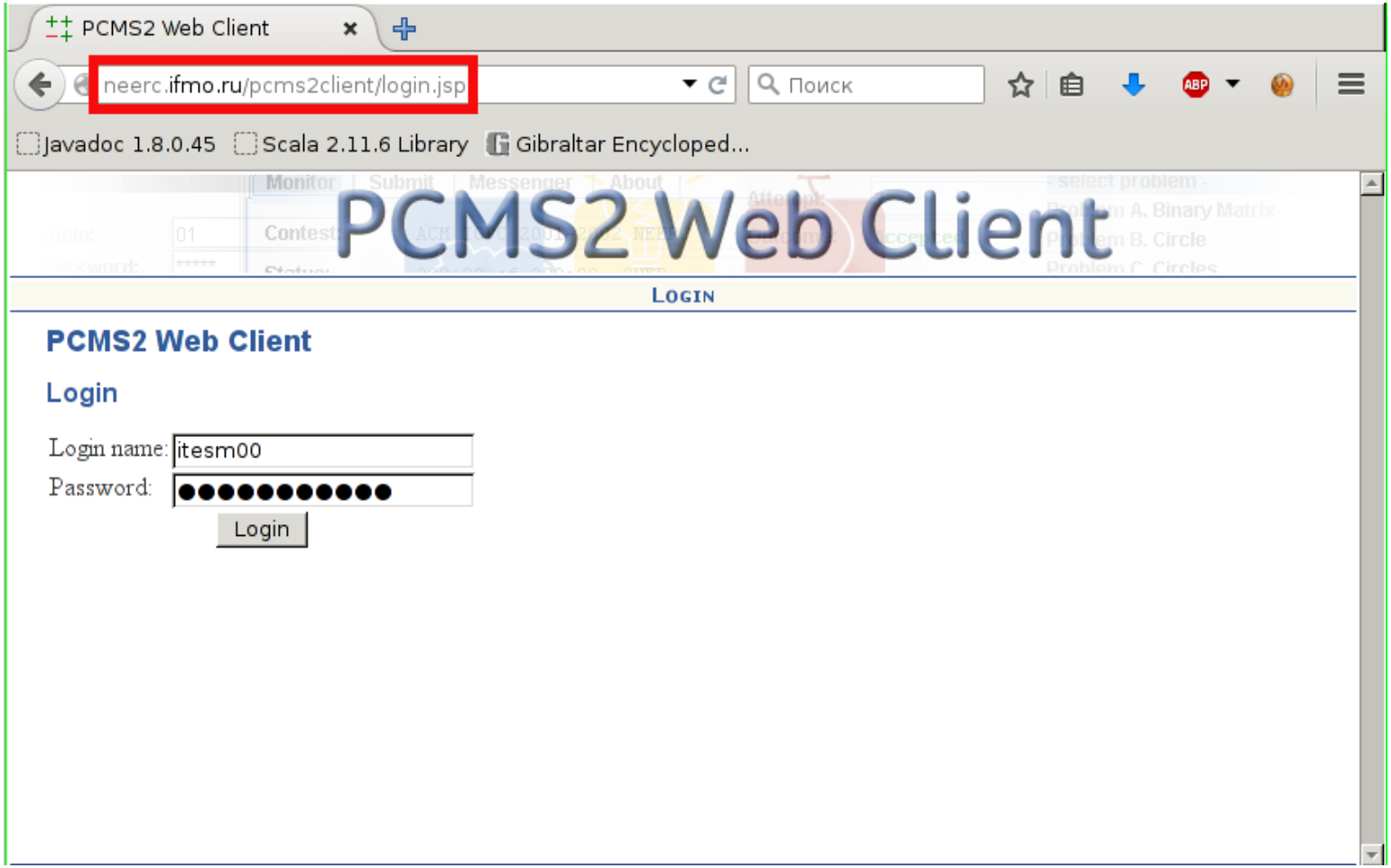
Common things for problems

- Testing machines: AMD Phenom II X4 955, 3.2GHz, 8G RAM
- Use files or stdin/stdout for input/output
  - typically `something.in`/`something.out`, if using files
  - "security violation" if waiting for `stdin`, if using files
  - be careful when reading problem statement
- Time limit: see problem statement
- Memory limit: typically 256 megabytes
- We report test numbers
  - "wrong answer, test 22"

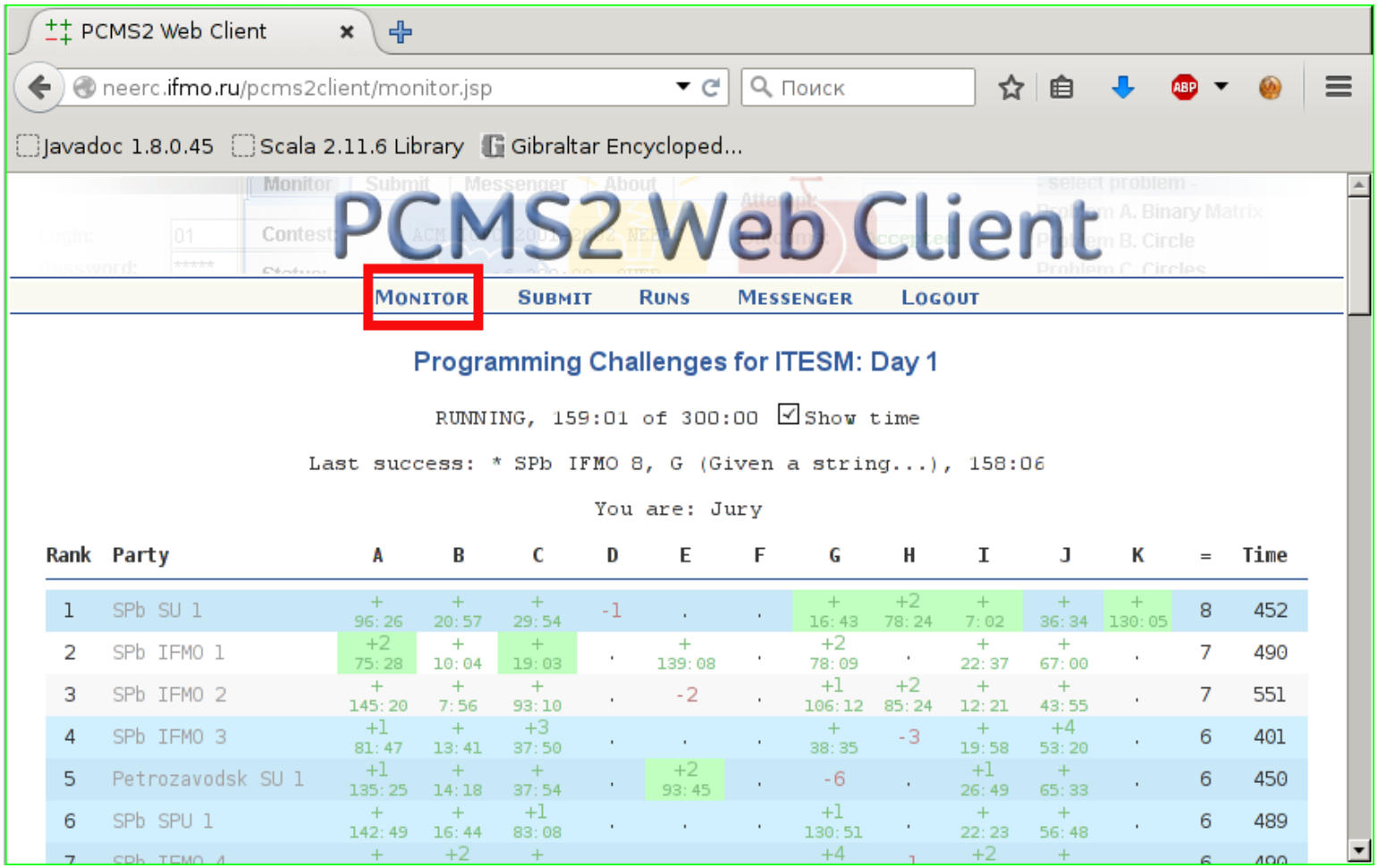
Common things for problems

- Testing machines: AMD Phenom II X4 955, 3.2GHz, 8G RAM
- Use files or stdin/stdout for input/output
  - typically `something.in`/`something.out`, if using files
  - "security violation" if waiting for `stdin`, if using files
  - be careful when reading problem statement
- Time limit: see problem statement
- Memory limit: typically 256 megabytes
- We report test numbers
  - "wrong answer, test 22"
- Usually first k tests are example tests from the problem statement

PCMS2 Client: Login window

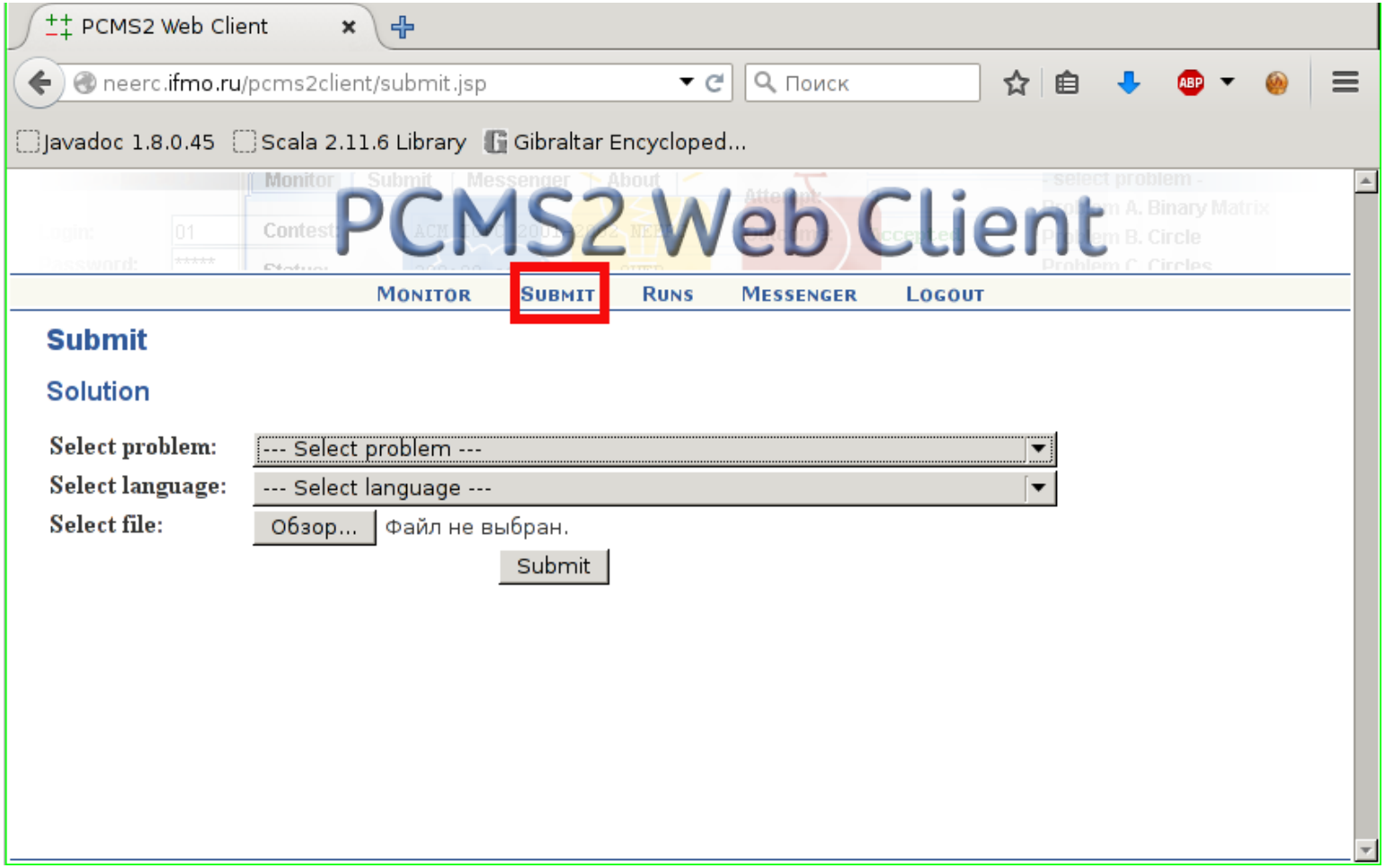


PCMS2 Client: Monitor

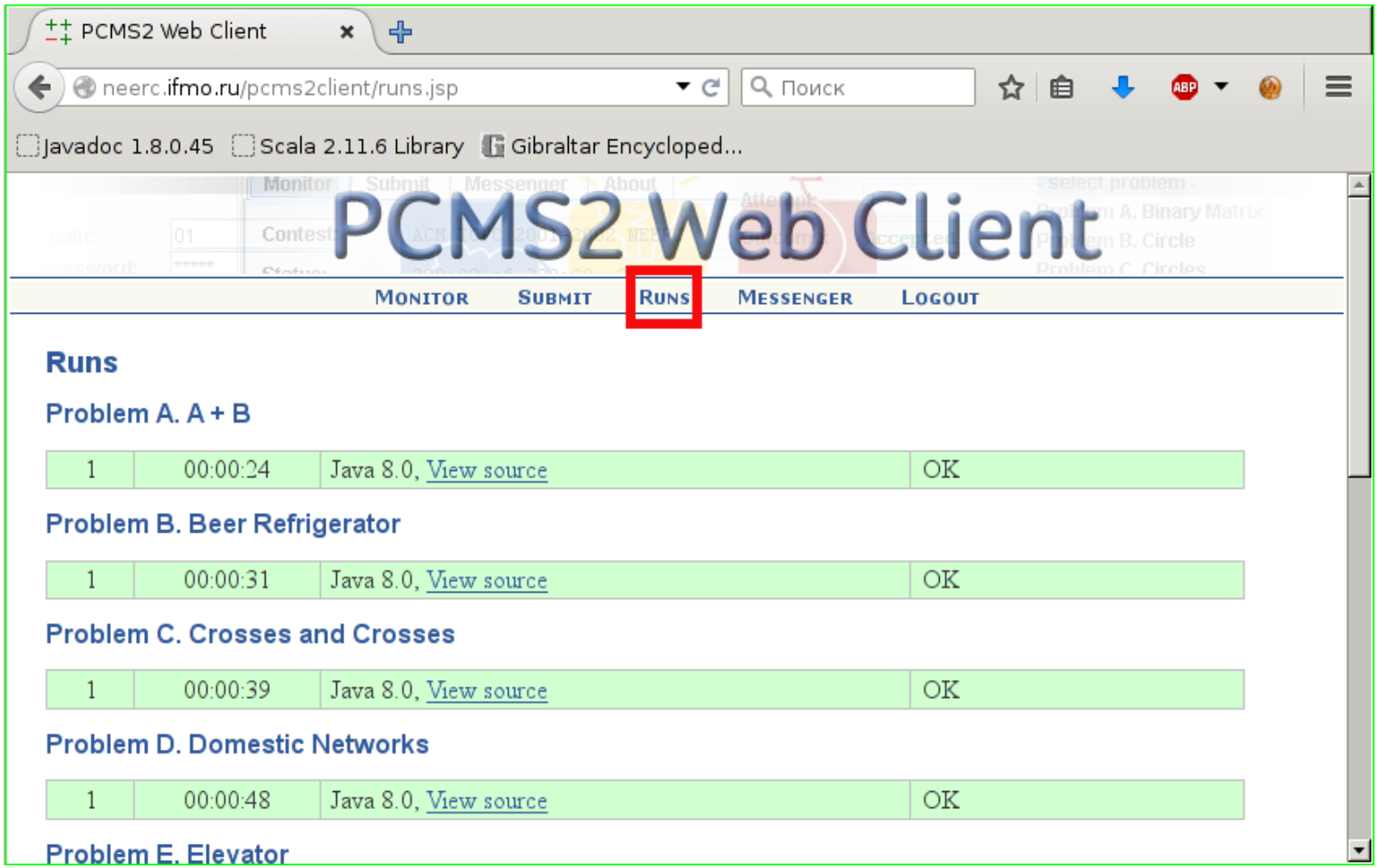


Rank	Party	A	B	C	D	E	F	G	H	I	J	K	=	Time
1	SPb SU 1	+ 96:26	+ 20:57	+ 29:54	-1	.	.	+ 16:43	+2 78:24	+ 7:02	+ 36:34	+ 130:05	8	452
2	SPb IFMO 1	+2 75:28	+ 10:04	+ 19:03	.	+ 139:08	.	+2 78:09	.	+ 22:37	+ 67:00	.	7	490
3	SPb IFMO 2	+ 145:20	+ 7:56	+ 93:10	.	-2	.	+1 106:12	+2 85:24	+ 12:21	+ 43:55	.	7	551
4	SPb IFMO 3	+1 81:47	+ 13:41	+3 37:50	.	.	.	+ 38:35	-3	+ 19:58	+4 53:20	.	6	401
5	Petrozavodsk SU 1	+1 135:25	+ 14:18	+ 37:54	.	+2 93:45	.	-6	.	+1 26:49	+ 65:33	.	6	450
6	SPb SPU 1	+ 142:49	+ 16:44	+1 83:08	.	.	.	+1 130:51	.	+ 22:23	+ 56:48	.	6	489

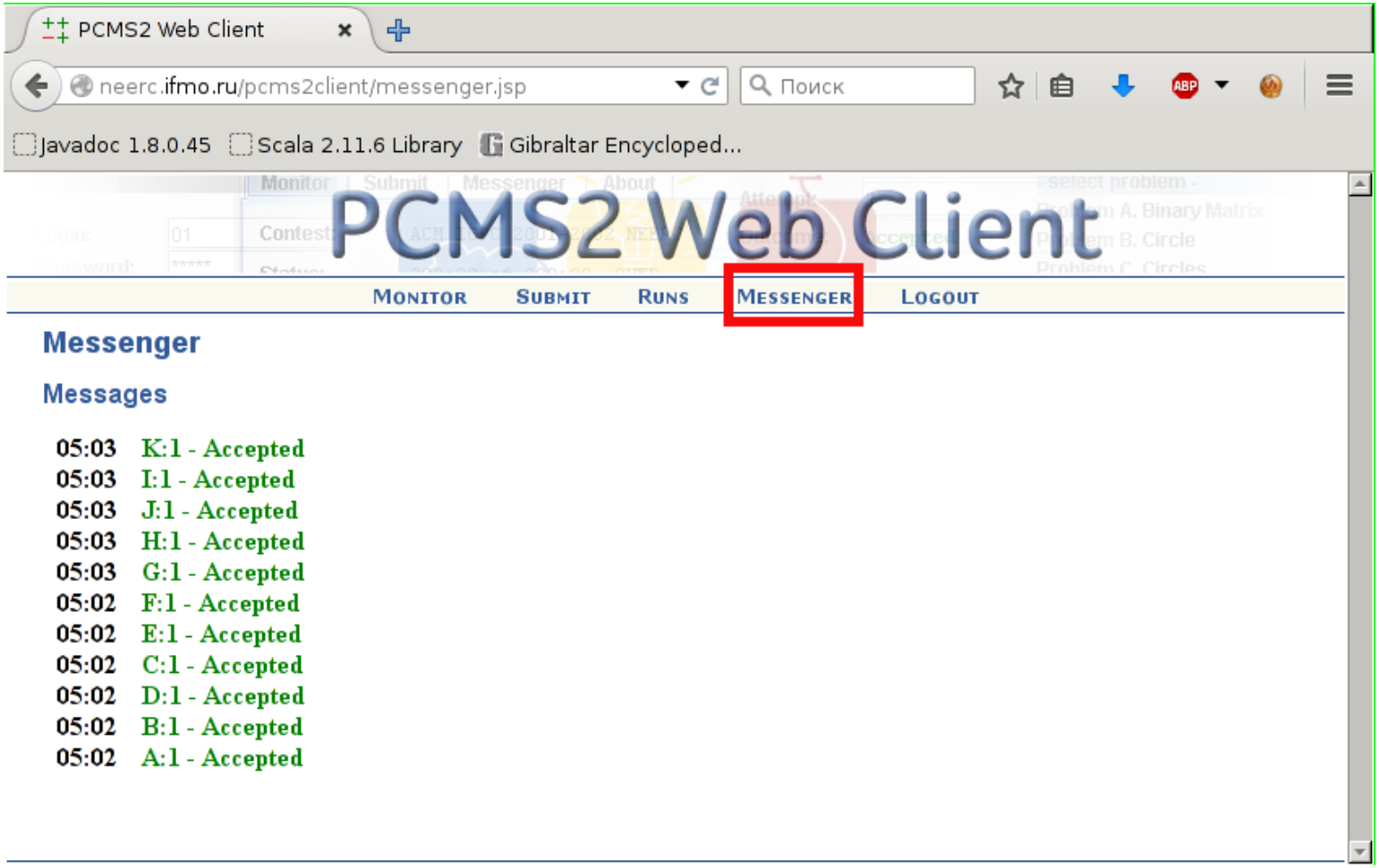
PCMS2 Client: Submit a problem



PCMS2 Client: View your runs



PCMS2 Client: View messages



Practice session

Practice session

- Get acquainted with the environment

Practice session

- Get acquainted with the environment
- We together solve two problems

Practice session

- Get acquainted with the environment
- We together solve two problems
- Then you solve two simple problems and one funny problem in an hour

Practice session

- Get acquainted with the environment
- We together solve two problems
- Then you solve two simple problems and one funny problem in an hour
  - in every language and every dialect you plan to use

Practice session

- Get acquainted with the environment
- We together solve two problems
- Then you solve two simple problems and one funny problem in an hour
  - in every language and every dialect you plan to use
  - C/C++: check how `std::cin`, `std::cout`, `printf`, `scanf` work with various data types (`long long`, `long double`, etc)

Practice session

- Get acquainted with the environment
- We together solve two problems
- Then you solve two simple problems and one funny problem in an hour
  - in every language and every dialect you plan to use
  - C/C++: check how `std::cin`, `std::cout`, `printf`, `scanf` work with various data types (`long long`, `long double`, etc)
- You may check system error messages, testing machine performance, etc