

CSES – Yet Another Online Judge

Antti LAAKSONEN, Topi TALVITIE

University of Helsinki, Department of Computer Science
e-mail: antti.laaksonen@helsinki.fi, topi.talvitie@helsinki.fi

Abstract. This paper describes the current state and future plans of the Code Submission Evaluation System (CSES) online judge project. Since 2013, CSES has been used to organize several online programming courses and contests in Finland, including algorithm courses at the University of Helsinki, the yearly Finnish Olympiad in Informatics (Datatahti), and the Baltic Olympiad in Informatics 2016. CSES is also known for the CSES Problem Set project whose purpose is to create a high quality problem collection for learning algorithmic problem solving, and also to document the history of programming problems.

Keywords: online judge, problem set, problem solving, competitive programming.

1. Introduction

The Code Submission Evaluation System (CSES) project started in 2013, and an early version of the system was used in a competitive programming camp organized at the University of Helsinki in summer 2013. The initial goal of the project was to establish a small system that could be used in local contests and training – in fact, many parts of the original system were implemented just the night before the first day of the camp.

In 2015, CSES was re-implemented from scratch and the domain `cse.fi` was registered. The new system was first used to organize an algorithm programming course at the University of Helsinki and the final round of the Finnish Olympiad in Informatics. However, the real ordeal for CSES occurred a year later when the Baltic Olympiad in Informatics 2016 was hosted by Finland and CSES was used to organize the contest.

During the first years, CSES was only used in Finland, but the situation changed in 2017 when the first version of the CSES Problem Set problem collection was published. Today CSES is an international system and most users come outside Finland. At the time of writing, the ten most active countries (in terms of the number of submissions) are Finland, India, United States, Russia, Vietnam, Croatia, Brazil, Morocco, Argentina and Bangladesh.

Recently, after releasing an updated version of the problem set, CSES has become a quite popular system in the competitive programming world. At the time of writing, the total number of submissions is about 700,000 and there are 5,000 new submissions every day.

2. System Overview

2.1. Features and Design

The two main features of CSES are courses and contests. A course consists of programming problems with automatic evaluation and text pages that can be used as online learning material. The most important course in the system is the CSES Problem Set; Fig. 1 shows an example problem statement in the course.

CSES supports both IOI style contests, where problems can have subtasks and partial scores, and ICPC style contests, where a submission is either accepted or not. CSES is

CSES Login —

CSES Problem Set
Sliding Median

TASK | STATISTICS

Time limit: 1.00 s **Memory limit:** 512 MB

You are given an array of n integers. Your task is to calculate the median of each window of k elements, from left to right.

The median is the middle element when the elements are sorted. If the number of elements is even, there are two possible medians and we assume that the median is the smaller of them.

Input

The first input line contains two integers n and k : the number of elements and the size of the window.

Then there are n integers x_1, x_2, \dots, x_n : the contents of the array.

Output

Print $n - k + 1$ values: the medians.

Constraints

- $1 \leq k \leq n \leq 2 \cdot 10^5$
- $1 \leq x_i \leq 10^9$

Example

Input:
8 3
2 4 3 5 8 1 2 1

Output:
3 4 5 5 2 1

Sorting and Searching

- ...
- Subarray Sums I
- Subarray Sums II
- Subarray Divisibility
- Array Division
- Sliding Median
- Sliding Cost
- Movie Festival II
- Maximum Subarray Sum II

Fig. 1. A problem in the CSES Problem Set.

#	name	score	A	B	C	D	E	F	G
1	pwild	670	100	100	100	100	100	100	70
2	ainta1	600	100	100	100	100	100	100	-
3	tutis	512	100	100	100	100	100	12	-
4	koosaga	500	100	100	100	100	100	-	-
5	zscoder	457	100	100	100	100	45	12	-
6	egor.lifar	457	100	100	100	100	45	12	-
7	Rudy358	457	100	100	100	100	45	12	-
8	bogdan10bos	445	100	100	100	100	45	-	-
9	faustaadp	412	100	100	100	100	0	12	-
10	toonewbie	412	100	100	100	100	0	12	-
11	SpeedOfMagic	412	100	100	100	100	0	12	-
12	DovydasVad	412	100	100	100	100	0	12	-
13	ppavic	400	100	100	100	100	-	-	-
14	jdnkjasska	400	100	100	100	100	0	-	-
15	ruhanhabib39	400	100	100	100	100	0	-	-
16	Diiven	400	100	100	100	100	-	-	-
17	cvele	400	100	100	100	100	-	-	-

Fig. 2. An IOI style scoreboard in the Datatähti Open 2019 contest.

used to organize the Finnish Olympiad in Informatics (Datatähti), and in recent years there have also been international versions of final rounds in the Datatähti Open contest series. As an example, Fig. 2 shows the scoreboard of Datatähti Open 2019.

While most problems in CSES are standard input/output tasks, CSES also supports other problem types, such as output only tasks, interactive tasks, library tasks and encoder/decoder tasks. CSES can also be used to organize virtual contests where each contestant has an individual time window. At the moment, CSES provides two virtual contest archives that contain all BOI and CEOI problems from 2005–2019.

Another feature, used in the CSES Problem Set, is hacking: after solving a problem, it is possible to view submissions from other users and try to find a test that breaks them. Fig. 3 shows the interface used in hacking. After a successful hacking attempt, the new test is automatically added to the test data and all submissions to the problem are re-evaluated.

To make sure that problems have correct test data and can be hacked, each problem has a validator that automatically checks the input format. In IOI style contests, validators are also used to determine which subtasks a test input belongs to.

The screenshot shows the CSES website interface for the 'Weird Algorithm' problem. At the top, there are navigation links: TASK, SUBMIT, RESULTS, STATISTICS, and HACKING. Below the navigation, there is a 'BACK TO HACKING' link and a table showing submission statistics:

time	victim	lang	max time
2020-04-25 11:32:08	ish	C++11	0.01 s

Below the table is a code editor showing the following C++ code:

```

1 #include <iostream>
2
3 using namespace std;
4
5 void weird_algorithm(long long n) {
6     if (n == 1) {
7         cout << n << "\n";
8     } else if (n % 2 == 0) {
9         cout << n << " ";
10        weird_algorithm(n/2);
11    } else {
12        cout << n << " ";
13        weird_algorithm(3*n + 1);
14    }
15 }
16
17 int main() {
18     long long n;

```

Below the code editor is an 'Input as text:' field containing the text '55555'. Below that is an 'Input as file:' section with a 'Browse...' button and the text 'No file selected.'.

On the right side of the page, there are two sections: 'Introductory Problems' and 'Your submissions'. The 'Introductory Problems' section lists several problems with checkboxes: Weird Algorithm (checked), Missing Number, Repetitions, Increasing Array, Permutations, Number Spiral, Two Knights, and Two Sets. The 'Your submissions' section lists several submissions with checkboxes: 2020-01-12 11:43:48 (checked), 2020-01-12 11:43:28 (checked), 2019-06-23 13:17:03 (checked), 2019-06-23 13:16:44 (checked), 2019-06-21 15:53:23 (checked), and 2019-06-20 17:43:32 (checked).

Fig. 3. In hacking, you can view other people's submissions and try to send a new test input that breaks the code.

2.2. Technical Details

CSES consists of two separate parts: the website and the judge servers. The website is implemented using PHP and PostgreSQL. The submissions are evaluated by a distributed cluster of judge servers. The judging environment is implemented in Rust, and to run the user-supplied code in a sandbox where it cannot harm the rest of the system, it uses the Isolate sandboxing utility (Mare , Blackham, 2012), which in turn uses the namespace and control group features of the Linux kernel.

To avoid timing interference between jobs, each judge server runs at most one job at a time. Because of this, a large number of servers is required, however they need not to be powerful in the parallel processing sense. The Intel NUC7i3 Mini-PC was found to be a cost-effective choice of hardware for this use case. To keep the cost down, the servers do not have internal mass storage drives; instead, the root filesystem is loaded into memory from a USB stick. The servers are upgraded by updating the filesystem images over the network.

3. CSES Problem Set

The CSES Problem Set is an ongoing project whose purpose is to create a high quality collection of educational algorithm programming problems. At the time of writing the problem set consists of 200 problems, and the final goal is to reach 1,000 problems.

Many problems in the problem set are *introductory problems* that teach problem solving techniques, such as how to use a certain data structure or how to apply an algorithm design idea. Here are examples of such problems:

- Given a set of coins, count the number of distinct ways you can create a sum when you can use each coin any number of times. (topic: dynamic programming).
- Create a data structure that efficiently supports the following operations: (1) update an array value, (2) calculate the sum of values in a range. (topic: segment tree).
- Count the number of permutations of $1, 2, \dots, n$ where each position i has some other value than i . (topic: combinatorics).

In addition, there are *advanced problems* that are more difficult and require more thinking. For example, such problems are:

- Given a grid whose each square is either black or white, efficiently find the largest rectangle where each square is white.
- Given a directed graph, add the minimum number of edges after which the graph is strongly connected.

There are some challenges in creating a large problem set: how to ensure the quality of the problems, how to classify the problems, and how to even keep track of which problems have already been added to the problem set. So far, we have mainly added problems that have been used in our courses and training camps over the years, but we expect that new tools will be needed for managing the problems in the future.

Another goal in the project is to document the history of programming problems: what were the first occasions when a technique was used, how it has evolved since then and how it is used now. For example, in the context of competitive programming, some techniques have become popular after appearing at an IOI, such as the centroid decomposition technique in 2011 and the two-dimensional segment tree in 2013.

4. Other Systems

There are a large number of online judge systems available that have various features and goals (Wasik, Antczak, Badura, Laskowski, Sternal, 2018). In this section we discuss other systems that have influenced our project.

The USACO training system (Kolstad, Piele, 2007) consists of programming problems and text pages, and it can be seen as a model for the CSES course feature. An important difference is that in USACO training, only a small number of problems is available at a time and you have to solve them all to proceed, while in CSES you can

solve any problems. The USACO approach clearly has pedagogical benefits, but it also restricts the use of the problems.

The UVa online judge (Revilla, Manzoor, Liu, 2008) is a pioneering system that provides a large number of problems from ICPC contests and other sources. The problems are divided into volumes and can be solved in any order. The UVa online judge problems can be used when reading the book *Programming Challenges* (Skiena, Revilla, 2003); in a similar way, the CSES Problem Set can be used when reading the book *Guide to Competitive Programming* (Laaksonen, 2017).

The CMS system (Maggiolo, Mascellani, 2012) has been used to organize recent IOI contests, and the Kattis system (Enström, Kreitz, Niemelä, Söderman, Kann, 2011) has been used to organize the ICPC World Finals and some regional contests. The goal in the development of CSES has been to support the features of those systems (problem types and scoring) and also to implement authentic scoreboards, so that both IOI and ICPC style contests can be practiced as virtual contests.

Codeforces (2020) is a popular competitive programming platform that features weekly programming contests and a problem set that consists of all problems used in contests. The hacking feature of CSES resembles that used in Codeforces contests. Another online judge that provides problem archives from IOI style contests is oj.uz (Problems, 2020). At the moment, it contains more problems than CSES; however, not all problem types are supported.

Why are there so many online judges? A natural reason is that everyone wants to have full control of their systems and make sure they can fix them and add new features when needed. Thus, it does not seem probable that the situation would change. Still, some more collaboration would be beneficial for the competitive programming community: at the moment it is difficult to move content from one system to another, because there are no common practices on how to represent problem statements, test cases, validators, etc.

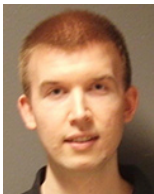
5. Acknowledgements

Mikko Sysikaski created many parts of the original CSES system. Roope Salmi has designed the current layout of CSES. Other contributors in the project are Matias Lahti and Henrik Lievonen.

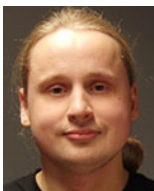
6. References

- Codeforces (2020). Access 2020-04. <https://codeforces.com/>
- Enström, E., Kreitz, G., Niemelä, F., Söderman, P., Kann, V. (2011). Five years with Kattis – using an automated assessment system in teaching. *IEEE Frontiers in Education Conference 2011*.
- Kolstad, R., Piele, D. (2007). USA Computing Olympiad (USACO). *Olympiads in Informatics*, 1, 105–111.
- Laaksonen, A. (2017). *Guide to Competitive Programming: Learning and Improving Algorithms Through Contests*. Springer.

- Maggiolo, S., Mascellani, G. (2012). Introducing CMS: A contest management system. *Olympiads in Informatics*, 6, 86–99.
- Mare, M., Blackham, B. (2012). A new contest sandbox. *Olympiads in Informatics*, 6, 100–109.
- Problems (2020). Access 2020-04. <https://oj.uz/>
- Revilla, M.A., Manzoor, S., Liu, R. (2008). Competitive learning in informatics: The UVa online judge experience. *Olympiads in Informatics*, 2, 131–148.
- Skiena, S.S., Revilla, M.A. (2003). *Programming Challenges: The Programming Contest Training Manual*. Springer
- Wasik, S., Antczak, M., Badura, J., Laskowski, A., Sternal, T. (2018). A survey on online judge systems and their applications. *ACM Computing Surveys*, 51(1), Article 3, 34 pages.



A. Laaksonen works as a university lecturer at the Department of Computer Science of the University of Helsinki. He is one of the organizers of the Finnish Olympiad in Informatics and has written a book on competitive programming. His role in the CSES project has been to implement the website and lead the creation of the CSES Problem Set.



T. Talvitie works as a postdoctoral researcher at the Department of Computer Science of the University of Helsinki. His main research interest is in optimizing algorithms to run efficiently on modern hardware. In the CSES project, his role is the implementation of the judging system and the