

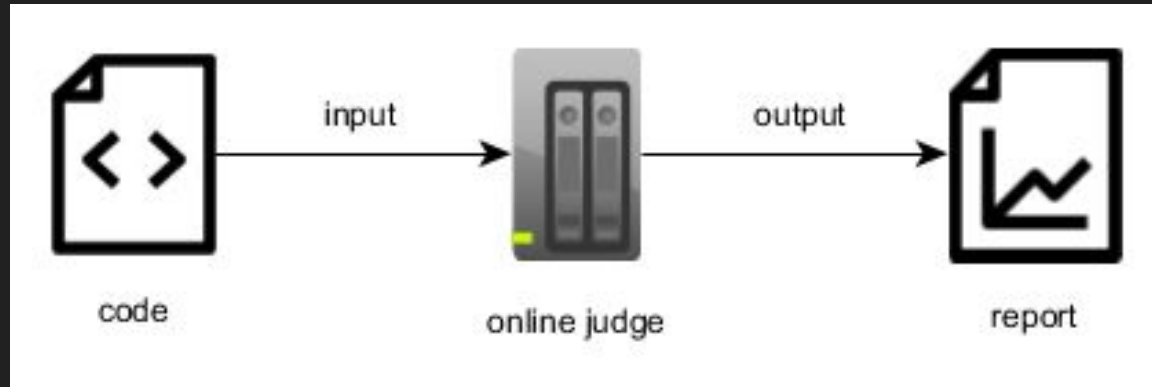
Design and implementation of the C++ online judge platform engine

Bachelor Diploma Thesis

Michał Leszczyński

Introduction

What is online judge?



purpose: algorithmic contests, teaching, recruitment of IT staff and more

task author



creates requirements

Requirements:

- > max 50% CPU
- > max 32 MB RAM
- > max 500 ms execution
- > test 1: result "YES" for input "aabaa"
- > test 2: result "NO" for input "dbabb"
- > test 3: result "YES" for input "abcba"

contestant



creates a solution

Contestant's program:

```
int main() {
    string s;
    cin >> s;
    int len = s.length();

    for (int i=1; i<len/2; i++) {
        if (s[i] != s[len-i-1]) {
            printf("NO\n");
            return 0;
        }
    }

    printf("YES\n");
    return 0;
}
```



online judge platform

Answer:

- > the program is compileable
- > test 1 was passed
- > test 2 was not passed
- > test 3 was passed



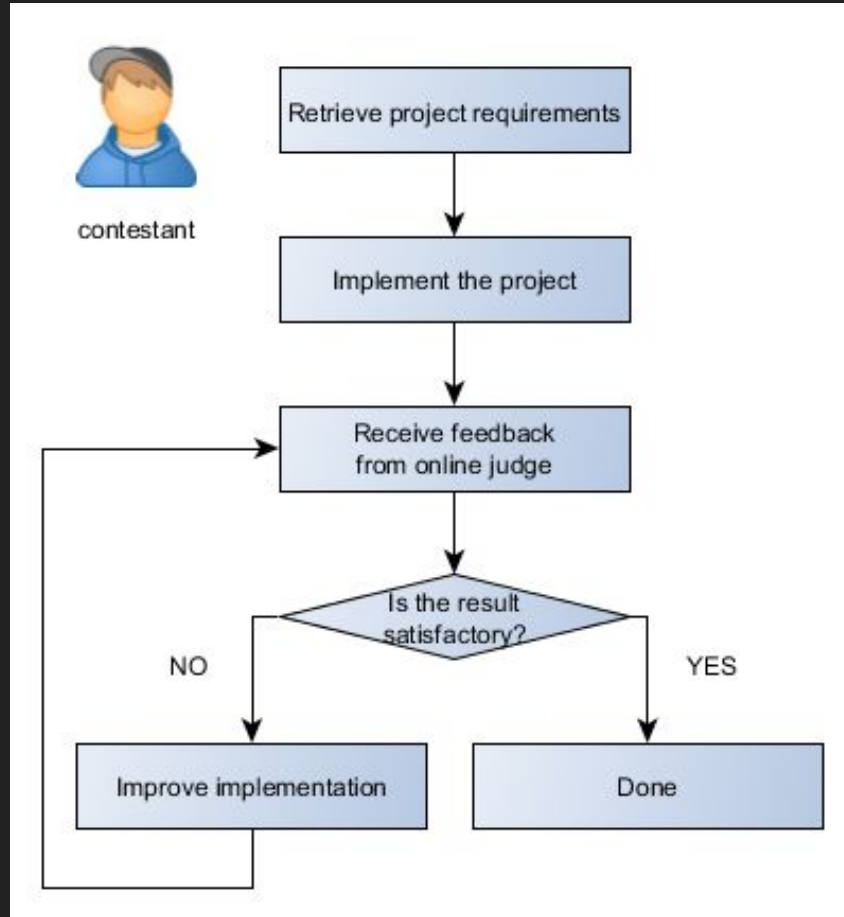
task author



contestant

Motivation

Feedback loop



Engine: differences in comparison to other solutions

- purely based on Python 3 and Docker
- Docker containers instead of Virtual Machines
- modular architecture, designed for easy maintenance
- described in detail in the thesis paper

Docker

- light containerization technology or Operating System Level Virtual Machine
- main difference to VMs: shared kernel
 - much easier to implement host-guest communication
- easy dependency provisioning (Docker Repositories)

Usage

Exemplary usage: performance tests

Suppose that contestant is provided with a task: “write a program which will efficiently sort N numbers”.

We will test three exemplary implementations:

- merge sort $O(n \log n)$
- bubble sort $O(n^2)$
- “random” sort $O((n+1)!)$

using payloads of length 6, 8, 10, 1024, 16383 with the time limit of 1.5 second and up to 50% single core CPU time.

Exemplary submission: random sort

Test name	Status	Points	Time [ms]	Memory [MB]
rand6	ok	1.0 / 1.0	2	0.668
rand8	ok	1.0 / 1.0	23	0.738
rand10	hard timeout	0.0 / 0.0	>= 1500	0.773
rand1024	hard timeout	0.0 / 0.0	>= 1500	0.633
rand16383	hard timeout	0.0 / 0.0	>= 1500	0.848

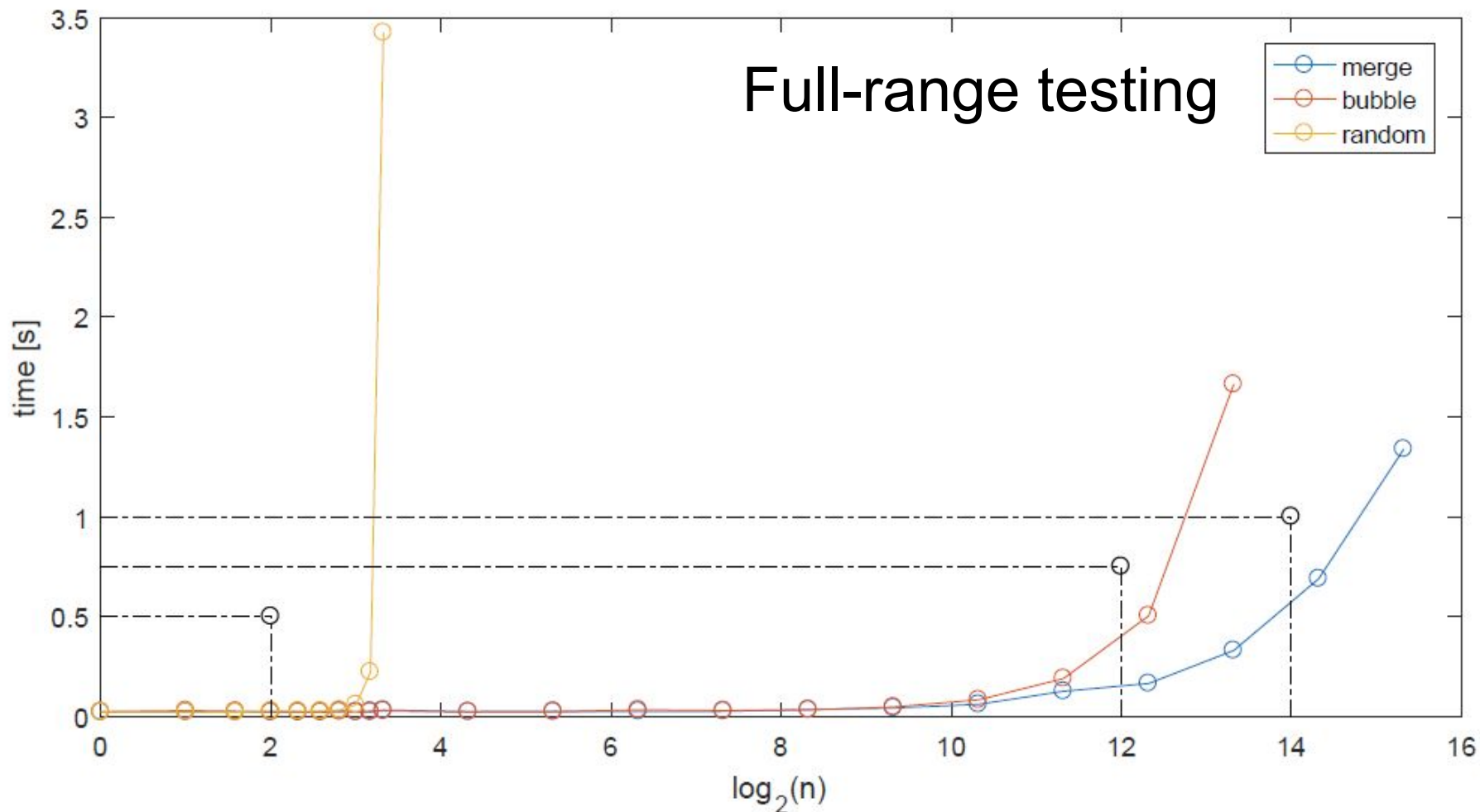
Exemplary submission: bubble sort

Test name	Status	Points	Time [ms]	Memory [MB]
rand6	ok	1.0 / 1.0	1	0.848
rand8	ok	1.0 / 1.0	2	0.688
rand10	ok	1.0 / 1.0	2	0.734
rand1024	ok	1.0 / 1.0	11	0.809
rand16383	hard timeout	0.0 / 0.0	≥ 1500	0.695

Exemplary submission: merge sort

Test name	Status	Points	Time [ms]	Memory [MB]
rand6	ok	1.0 / 1.0	2	0.719
rand8	ok	1.0 / 1.0	2	0.781
rand10	ok	1.0 / 1.0	2	0.699
rand1024	ok	1.0 / 1.0	3	0.793
rand16383	ok	1.0 / 1.0	33	0.902

Full-range testing



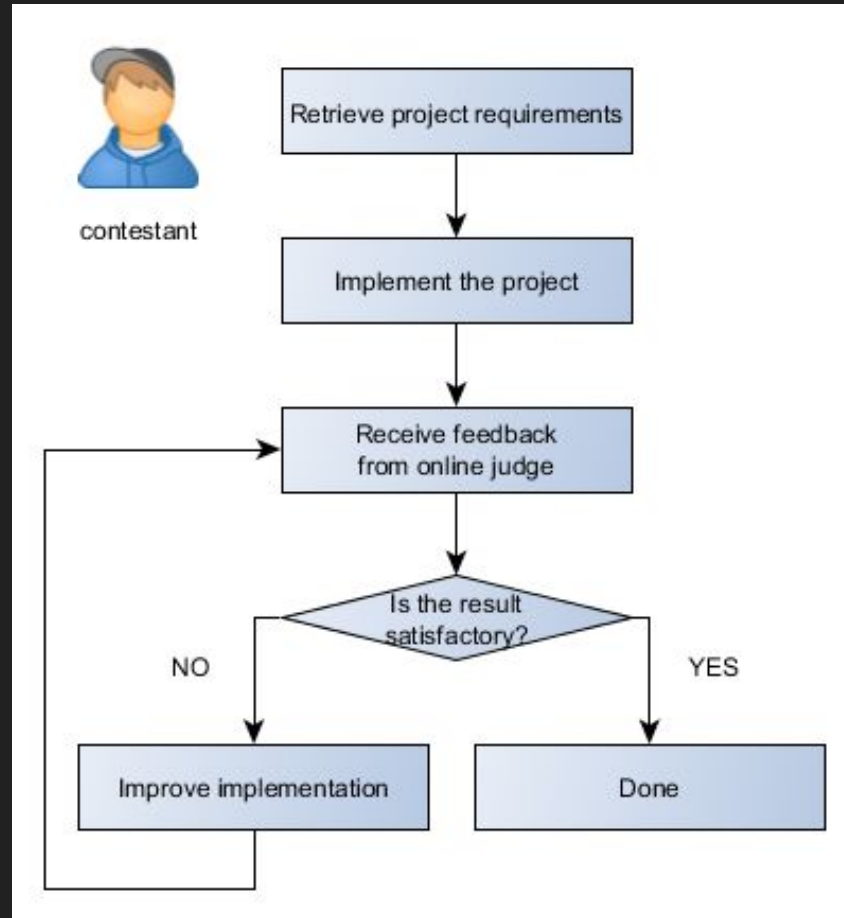
Applications

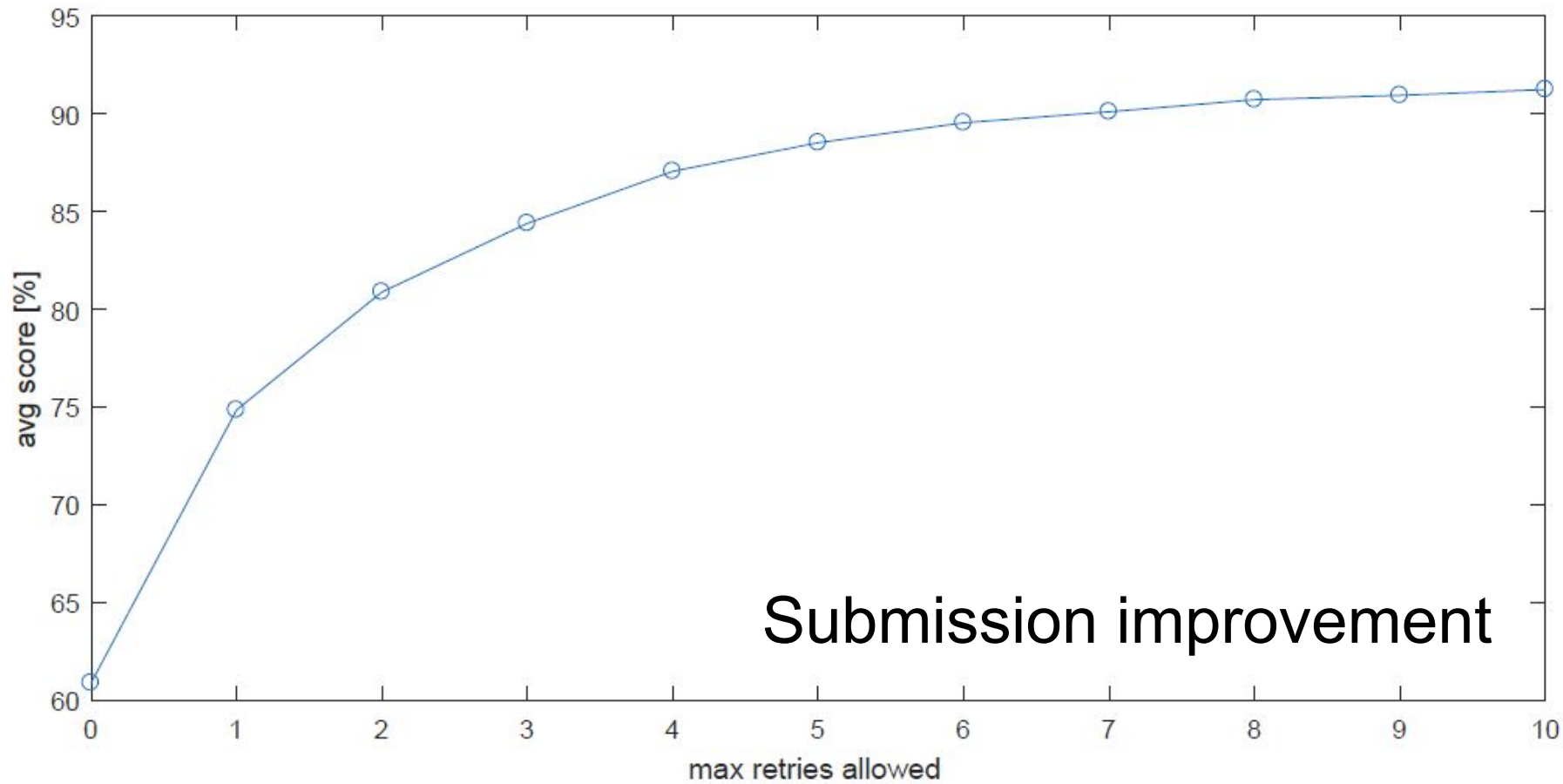
Practical applications (on FEIT WUT)

- deployed under algo.elka.pw.edu.pl
- algorithmic contests
 - faculty qualifications for CS olympiads
- teaching
 - continuously used as a teaching aid on Programming Fundamentals lecture since semester 16L
 - up to now:
 - 322 registered users,
 - 208 tasks,
 - 8762 submitted projects.

Evaluation

Back to the
feedback loop





Submission improvement

Anagram

Description

Prepare program that checks whether two given strings are anagram. Strings are separated with new line. Print YES in case of anagram or print NO otherwise. Use arrays.

Example

Input:

```
opsss  
pssss
```

Output:

```
YES
```

Submissions 1

Submit time	Status	Result	Message	
26.08.2018, 11:01:52	✓ Accepted	100%	Perfect Work! Well done	View report

[Submit new solution](#)

Deadline: 26.07.2018, 10:00:00

Before submitting your code, please read the
carefully

Thank you for your attention
Questions?