

SIST オンラインジャッジシステムの授業における運用報告

Operation report of SIST online judgement system for introductory education of programming

山下 颯也*、高田 洸*、國持 良行*
Soya YAMASHITA, Hiro TAKADA and Yoshiyuki KUNIMUCHI

We hosted a physical sever, named "SOJ", outside the university and put it into operation, in order to utilize it for introductory education of programming. Then, it was tested and operated in the compulsory subject "Programming Exercise" in the second semester of 2021. As a result, it contributed to the home study time. Two ability tests were conducted to measure the effect of SOJ. But the learning effect of SOJ could not be sufficiently verified due to lack of data.

1. はじめに

競技プログラミング等でよく使われるオンラインジャッジシステム(Online Judgement System, 以下 OJS)をプログラミングの導入教育に活用するために, 学外に専用サーバをホスティングし, そこへ OJS システムを構築した(以下, SOJ とよぶ). そして, 2021 年度 1 年後期必修科目「プログラミング演習」において試験運用した. SOJ の効果を測るために 2 回の実力テストを実施して, SOJ の使用状況や学習効果を検証した.

我々は, 昨年度家庭学習で SOJ を活用してプログラミング能力を向上させることを目的として, 学内ネットワーク上に構築した SOJ を, 学外へ移行する計画を立てた¹⁾. 本年度はその計画を遂行し, サーバの移行と試験運用をすることができた. 当初の計画よりも遅延が発生して, 試験運用には十分な期間が得られず, 当初予定していた学習効果を十分に検証することはできなかった.

会津大学の AOJ²⁾, 北京大学の POJ, AtCoder などがよく知られており, ランキング, コンテストやリクルートなどで活用されている. Moodle のプラグインとしての OJS なども提案されている⁴⁾ が, サーバに負荷をかけ過ぎる不安がある.

また, OJS をプログラミング教育に活用する報告もみられる^{3) 5) 6)}. 不正コピー防止^{7) 8)} や作問の効率向上⁹⁾ についての提案もみられる.

本論文において, 試作した SOJ の使い方(外部仕様)を 2 章で, 専用サーバ上での構築(内部仕様)を 3 章で, 「プログラミング演習」で運用した結果を 4 章でそれぞれ紹介する.

2. OJS の概要と SOJ の外部設計

この節では, 一般的な OJS の概要を紹介し, それに沿っ

て SOJ の外部仕様を紹介する.

2.1 OJS とは

OJS は, 一般に Web 上のサーバが学習者から提出されたプログラムを自動採点し, その結果を DB へ記録, 学習者へフィードバックするシステムである.

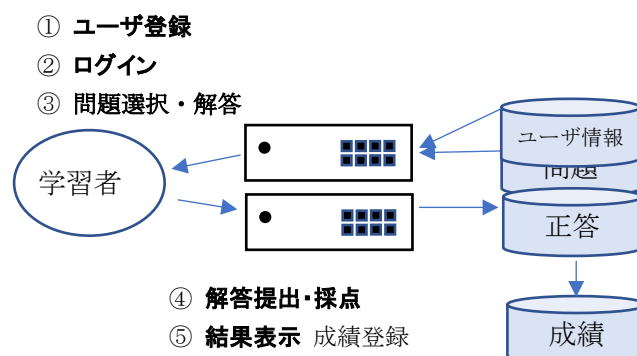


図 2.1 OJS での学習方式

学習者は, 初回のみ① ユーザ登録をする. まず, ② システムにログインする. ③ 用意された問題の中から 1 つの問題を選択して, プログラムを作成する. 学習者は, プログラムを作成し終えたら, ④ そのソースコードを提出欄に記述して提出(submit)する. 提出後, システムは自動採点を行い, ⑤ 学習者へ結果を返して, データベースに問題番号, 正解不正解の状態, 実行時間, 最終日時などを登録する.

上記の学習過程 ③～⑤を繰り返し, 成績は DB へ蓄積される. 成績記録から, 学習者のプログラミング能力, モチベーション, 全体成績, 問題難易度などを可視化し, 学習者の学習意欲を引き出す.

2.2 システムの外部設計

本システムに携わる人(利用者)を、学習者、教師、管理者の3つに分類する。システムには最初ただ一人の管理者 root/admin が存在し、ユーザ登録・削除などの特権が与えられる。教師と学習者をそれぞれ1名以上登録していると仮定する。そして、以下では利用者が学習する際の画面を紹介しながら、外部仕様について述べる。

2.2.1 学習ルーチン(学習者モード)

(アカウントをもつ)学習者は SOJ にログインし、問題解答、問題作成、結果閲覧などを行うことができる。

学習者のログイン

SOJ の URL は、以下の通りである。ただし、現在 Basic 認証(ID と PW)を設定している。

<https://sist-onlinejudge.com>

なお、このサイトは、さくらインターネットからホスティング専用サーバ上に構築されており、IPv4 アドレスも1つ付与されている。ログイン認証後、SOJ のトップ画面が表示される(図 2.2)。

以下では、①初回のみ必要なユーザ登録、②ログイン、③問題選択・解答、④解答提出、⑤ 採点・成績表示の手続きを紹介する。



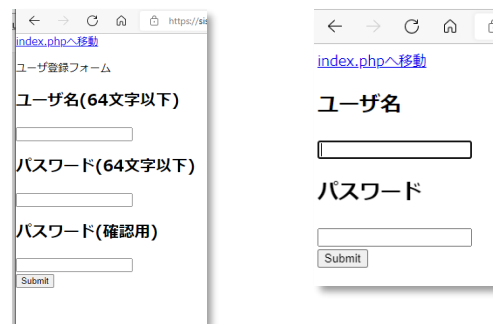
図 2.2 SOJ のトップ画面

① ユーザ登録

最初に SOJ を利用する際には、ユーザ登録をしなくてはならない。右上付近のメニュー(図 2.2 の矢印部分)から「ユーザ登録」を選択する。ここでは、ユーザ登録画面(図 2.3①)にて ID とパスワードを決定する。

② ログイン

右上付近のメニューから「ログイン」を選択する。ログイン画面(図 2.3 ②)にて①で設定した ID とパスワードを入力する。



① ユーザ登録画面 ② ログイン画面

図 2.3 ユーザ登録画面(左)とログイン画面(右)

③ 問題選択・解答

右上付近のメニューから「問題一覧」を選択する。問題一覧ページ(図 2.4)へ移動したら、問題を選択して、問題の解答に入る。問題番号と問題文、サンプルの入力とそれに対する出力の例がいくつか記述されている(図 2.5)。学習者はエディタやコンパイラを使ってプログラムを作成する。

問題ID	問題名	提出状況
1	HelloWorld	正解
2	AxB	正解
5	XX	未正解
6	testtett	未正解
7	POW	正解
8	Q0052ITLCM	正解
10	Q0053ARBackOrder	正解
11	Q0054_IT_SUM	正解
13	Q0051_OP_divisor	正解
14	Q0058_OP_RemainderMult	正解
15	Q0059_IT_Factorial	正解
16	Q0057_IT_CenRem	正解
19	Q0019_CD_NaturalNumberComparison	正解
21	Q0021_IT_SquareDrawing	正解

図 2.4 ③問題一覧画面

【現状の課題】プログラミングコンテストでは**制限時間**が設けられていることが多いが、本システムでは学習を対象としているので、プログラムの作成に制限時間を設けていない。また、個人成績一覧から正解していない問題を選ぶのが一般的である。学習者のレベルに応じた問題や苦手分野の問題などをシステムが AI や DS を利用して**自動出題**を可能としたい。現在 C/C++言語のプログラムのみを採点できるが、将来的には**他言語への拡張**(Python や Java など)を図りたい。

問題

二つの整数a,bが空白区切りで与えられるので、a*bの値を改行込みで出力してください。

制約

$1 \leq a, b \leq 10000$

入出力例

入力1

5 6

出力1

30

入力2

318 2942

出力2

1818156

Submit

図 2.5 SOJ の問題出題画面

④ 解答(プログラム)提出・採点

問題 ID と問題名を選択し、作成したプログラムをコピーして、貼り付ける(図 2.6). そして、「Submit」ボタンを押す。

Input SourceCode

1 HelloWorld

```
#include<stdio.h>
int main()
{
    puts("Hello World");
    return(0);
}
```

Submit

図 2.6 SOJ の解答提出画面

【現状の課題】1つの解答の採点には1分程度かかる。この時間は、1つの問題に10個のテスト入力を用意されていて、1個のテスト入力に対する判定に余裕をもって約5秒かけていることによる。採点の結果は1分程度後に次の⑤結果表示のページに移動したときに初めて知ることができる。**採点中の採点状況**を表示し、採点終了後には採点結果を表示する機能が欲しいところである。

また、自動採点には他人のプログラムの**コピー&ペースト**が問題となる。コピペ対策を論じている先行研究もある。我々は、字句解析ツール(flex など)を使って**キーワードの出現分布**をコピペ発見に活用することを考えている。

⑤ 結果表示(結果一覧)

SOJ は解答(プログラム)を採点して、以下の結果を判定する。

- ・ 正解(Accepted, AC または Correct Answer, CA)
- ・ 不正解(Wrong Answer, WA)
- ・ コンパイルエラー(Compile Error, CE)

- ・ 実行時エラー(Runtime Error, RE)
- ・ 制限時間超過(Time Limit Exceed, TLE)
- ・ 記憶制限超過(Memory Limit Exceed, MLE)

問題ID	最終結果	実行時間	メモリ使用量	
1	AC	0	0	view
2	AC	0	0	view
5	WA	0	0	view
5	WA	0	0	view
1	WA	0	0	view
1	AC	0	0	view
1	AC	0	0	view
8	AC	0	0	view
11	AC	0	0	view
13	AC	0	0	view
7	AC	0	0	view

図 2.7 ⑤結果表示画面

【現状の課題】TLE や MLE は試作版では実装していないが、time.ps コマンドなどを使って今後実装を図りたい。

2.2.2 問題作成(学習者モード)

右上のメニューから「問題作成」を選択すると、問題作成画面(図 2.8)が表示される。問題制作者は、問題の名前を与え、その後 html ファイル、10 個の入力データファイルと 10 個の正答データファイルを選択し、提出ボタンを押す。

問題作成

問題名

問題文ファイル
ファイルの選択 ファイルが選択されていません

入出力の上限は各30までです。
入力ファイルの数と出力ファイルの数は一致している必要があります。

入力ファイル
ファイルの選択 ファイルが選択されていません

出力ファイル
ファイルの選択 ファイルが選択されていません

提出

図 2.8 問題作成画面

OJS の採点は、提出されたプログラムに対して、テスト入力を読み込ませ、プログラムからの出力を記憶する。その出力を、あらかじめ用意されたサンプル出力と比較する。すべてのテスト入力に対する出力がサンプル出力と一致し、かつ実行時間が制限時間内である場合に、正答 AC を与えるものとする。すなわち一種のブラックボックステストである。したがって、厳密な意味で正しいプログラムを証明するものではない。しかし、実用上の問題はない。

作成する問題の要件としては、入力を読み込んで、結果を出力する仕様にする必要がある。テスト入力に関しては、典型的な入力、境界値近辺のデータ、サイズが上限近辺のデータなどを作る必要がある。

そのためにはあらかじめ模範解答のプログラムを作成しなくてはならない。入力ファイルをいくつか用意する。ファイルはこれらから導出できる機能を用意しておくことで作業効率が上がる。

【現状の課題】「プログラミング演習」を対象にした良問を作成する。数学の中学入試には参考になる良問が多い。また、作問の作業は、3年前期「コンピュータシステム実践演習1」において、受講生に問題とテスト入力と正解出力を作成してもらう予定である。昨年の反省点としては、問題を互いに異なる内容にすることである。あらかじめ学生にこういうテーマの問題を作成するように指示する必要がある。また、来年度にはSOJが稼働しているので、制作した問題の確認も行うことができる。

2.3 システムの外部設計のその他の課題

外部仕様においてまだ実現されていない課題について言及する。

2.3.1 CSSやJavaスクリプトによるWebデザイン

本年度稼働したSOJは、試験運用のため動作確認が主目標であった。従って、Webページは最低限必要なデザインや機能しか付加していない。まず、Webページのデザインを、CSSやJavaスクリプトを使って見やすいものに作り込むことがあげられる。次に、学習者にアンケートした結果からは以下のような使い勝手に関する意見があった。

- ・問題一覧で1ページ当たりの問題数の上限を付ける。
- ・問題選択画面にて問題難易度の情報がほしい。
- ・問題を分野別に出题する。
- ・問題を問題名で検索する。
- ・問題文と提出欄を同じページに配置する。
- ・提出欄はもっと広い方がよい。
- ・システムが採点中に詳細な経過が分かるようにする。
- ・採点が終了したときに採点結果を学習者に提示する。
- ・自分の成績表を一覧でき、自分の実力を視覚化する。
- ・利用者全体のランキング(上位数名)を表示する。

などである。

2.3.2 教師モードと管理者モードの追加

文献[1]では、学習者の集団を担当する教師、システムの運用を行う管理者の利用を想定している。そのためのアカウントと機能を付加する予定であったが、今回はその機能は実現されていない。

SOJでの学習状況を教師が把握し、数値化するために、以下のことを実現する。クラス編成(担任学生の設定)、クラスの成績サマリ表示(およびダウンロード)

3. システム概要

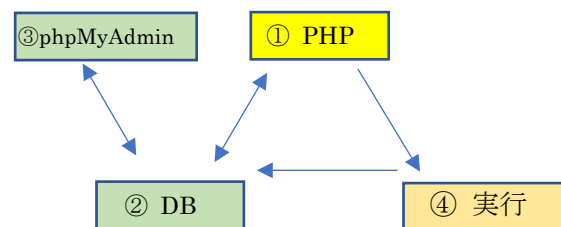
ここではSOJのシステムの概要を述べる。

3.1 システム概要

本SOJは、本システムは、主に①インターフェースを担当する**PHP コンテナ**、②ユーザ情報、管理情報、問題情報を管理する**DB コンテナ**、③DBとのインターフェースを担う**phpMyAdmin コンテナ**、④プログラムを起動し、正誤を判定する**実行コンテナ**からなる(図3.1)。

① PHP		③ phpMyAdmin	g++,
Apache(Web)	③ DB(mysql)	Apache(Web)	④ 実行
コンテナ	コンテナ	コンテナ	コンテナ
Docker			
リアル OS			

(a) システム動作環境



(b) システムの処理の流れ

図 3.1 システム概要図

①～④は**Docker**の仮想環境で動作するコンテナとして実現されている。コンテナは、仮想のマシン/OS環境であり、特定の環境下でソフトウェアを動作させることを可能にする。したがって、1台のリアルマシンによって、複数台のマシン上で動くソフトウェアが協調するシステムを構成することができる。

① **PHP コンテナ**は、Webサーバ上で動作し、利用者の対応(front end)を担当する。ユーザからは、プロフィール設定(登録・削除・変更・プロフィール確認)や操作要求(ログイン・ログアウト・問題選択・採点・成績表示・履歴おいて表示)などの要求を処理する。

② **DB コンテナ**は、ユーザ情報、管理情報、問題情報などを管理する。**ユーザ情報**はプロフィール、提出履歴、成績などが含まれる。**管理情報**には、クラス編成、コンテストの情報などがある。**問題情報**には、問題集とその各問題に対するテストデータが含まれる。問題には学習分野と難易度を付けたい。問題は、学習者(ユーザ)に理解できる言語(日本語や英語)で書かれ、解答されるプログラム言語に依存しないことが望ましい。

③ **実行コンテナ**は、学習者(ユーザ)が解答(プログラム)提出時に選択した言語処理系へ渡して、実行結果と正答デ

ータと比較して、正誤を判定する。現在選択できる言語処理系は C/C++コンパイラのみであるが、順次言語処理系を増やしていく予定である。

正答データは、テスト入力(ファイル)とそれに対する正しい出力が格納されたファイルの複数の集まりである。すべてのテスト入力に対して、提出プログラムが正しい結果を出力した場合に、正解と判定する。テスト入力に対して制限時間内に実行が終わらなければ、その実行を Docker コンテナ(内の OS)が強制的に打ち切り、「制限時間オーバー」(TLE)を判定する。また、実行中にエラー(例外など)が発生したときにも、強制的にプロセスを停止し、「実行時エラー」(RE)と判定する。これらの結果は DB の提出履歴と成績に反映される。

悪意のあるプログラムが入力される恐れがあるので実行コンテナの実行権限、ファイルアクセス権限は最低に設定する。

3.2 システムの実行環境

SOJ のシステム構成は、2022/2/10 現在以下の通りである。

【ハードウェア・OS】

OS: Ubuntu 20. 04

【ソフトウェア】

Docker(プロセス管理): 20.10.8

Apache(Web サーバ): 7.2

PHP(インターフェース): 7.2.34

MySQL(データベース): 8.0.26 - MySQL Community Server - GPL

GCC(コンパイラ): 8.3.0

3.2 Docker による実装

Docker による実装方法を述べる。

3.2.1 Docker とは

Docker とは、コンテナ仮想化によってアプリケーションを開発・実行するためのオープンソースソフトウェアである。この機能を用いることによって、簡単かつ高速にコンテナを立ち上げることができ、コンテナを分散することによって、意図せぬ動作による不具合の影響も軽微に抑えることが可能になる。

Docker-compose とは、1 つのアプリケーションを複数のコンテナとして組み合わせて定義するファイルである。これを用いて、PHP コンテナ、DB コンテナ、実行コンテナを合わせて1つの SOJ アプリケーションとして構築されている。

3.2.2 提出プログラムの採点待ちと割り付け

PHP コンテナに提出されたプログラムは、まず実行コンテナに送られる。実行コンテナに送られた後は、拡張コマンド task-spooler などを用いてキューに登録し、実行用のプロセスが空き次第実行に移す。採点の実行後は、採点の結果をデータベースに格納しプロセスを終了する。

現在は実行用のコンテナが 1 つであり直列での処理になっ

ているが、ゆくゆくは 3~5 個程度の実行コンテナを用いて並列処理を行う予定である。コンテナの状況を見て割り振りを行う必要があり、課題である。

3.3 データベースの構成

データベースのテーブルのうち主なものは、ユーザ管理用テーブル(表 3.1)、問題管理用テーブル(表 3.2)、提出結果管理用テーブル(表 3.3)の 3 つである。提出結果においては、大量の提出によるデータベースの容量不足が危惧されるため、user と problem の組み合わせに対して、正答かつ最新の提出プログラムだけを保持するものとしておく。

表 3.1 ユーザ管理用テーブル

列名	データ型	説明
id	int	[主キー]ユーザの id
name	varchar(64)	ユーザ名
auth	int	ユーザの持つ権限
pass	varchar(256)	パスワード(ハッシュ化)

[Auth について] DB のアクセス権限は以下の順である。

0 → 管理者, 1 → 教師, 2 → 学習者(学生)

表 3.2 問題管理用テーブル

列名	データ型	説明
id	int	[主キー]問題の id
user	int	問題作成ユーザの id
title	varchar(64)	問題名
time	int	実行時間制限(秒)
memory	int	実行メモリ制限(MB)

表 3.3 提出結果管理用テーブル

列名	データ型	説明
id	Int	[主キー]提出結果の id
user	Int	提出ユーザの id
problem	Int	提出問題の id
lang	Int	言語
status	varchar(16)	実行結果
time	Int	実行時間
memory	Int	使用メモリ
timestamp	Date	提出日時
path	varchar(256)	正答の格納パス

4. 試験運用とその結果

現在、SOJ を試作し、基本的な動作を確認している。今後の運用に向けての計画を述べる。また、運用中の課題をシステムへフィードバックする

4.1 外部サーバへの移植過程

2021 年 7 月ごろまでには、SOJ は学内のローカルサーバ上でほぼ現状と同じ程度に開発されており、後期からの運用に向けて外部のサイトへ移植することが課題となっていた。移植の過程を以下に述べる。

前期のコンピュータシステム実践演習 1 にて 100 題弱からなる問題集を作成した。夏期休業中には、移植手続きの立案と来年度以降の開発・管理・運用者の育成のための打ち合わ

せを10数回開いた。

8/25に、さくらインターネットの専用サーバPHYを1台レンタルする契約を結んだ。ハードウェア仕様は、

モデル名:RX1330 M3 4コア 1CPU

CPU: E3-1220 v6 3GHz (4Core)

メモリ: 8GB

ストレージ1: SSD 480GB x2

である。まず、さくらインターネットの「コントロールパネル」ページから専用サーバへ、以下のようにOSのインストールし、rootやネットワークの設定をした。

OS: ubuntu20.04

IPv4 アドレス: 153.126.242.45

そして、専用サーバでの設定・開発・運用を安全かつ高速に行うために、SSH 通信経由でコマンド実行している。そのためSSHのユーザ設定などを行った。

次に、Docker をインストールして、仮想サーバに切り分けた。その後仮想サーバへApache, PHP, MySQL, GCCなどのソフトウェアをインストールした。Dockerを用いた利点は、将来システムを拡張する際に、処理を並列化して大量のトランザクションに対応可能となることである。

9~10月にはDNS名の取得、サーバ証明書の取得、SSL通信を実施した。

サーバURL: <https://sist-onlinejudge.com>

なお、セキュリティのため、Basic認証がかけられている。

運用にあわせてシステムを微修正して、簡単なマニュアルを用意した。そして、11/23のプログラミング演習の第10回目の授業にて、1年生へSOJの紹介を行った。課題は、本学の内部と外部の間にあるファイアウォールによって、ポートが閉鎖されていて、学内ネットワークからSOJサーバへアクセスができない点である(現在対策中)。

4.2 試験運用

情報学部1年生後期必修「プログラミング演習(国持クラス)」(火曜日2時限目、対面実施、履修者数23名)において、SOJを試験的に運用した。10回目の授業に当たる2021/11/23(火)2時限目にSOJの目的・操作手順・成績閲覧の方法を紹介し、問題一覧から問題を選択して、家庭学習にて解答するように指示をした。第13回目12/14には、再度SOJを利用するためのリマインドを実施した。そして、第14回目12/21に第1回実力テストを、第15回目1/11に第2回実力テストをそれぞれ実施した。

4.3 SOJの利用状況

11/21(火)の第10回目の「プログラミング演習」でSOJのアカウント作成、操作法の紹介をした。1年生はこの日に使用が始まっている。そして、定期試験前日の1/24まで63日間の利用状況について考察する。まず、総提出件数は536件であった。その内訳は、

正解(AC)の数: 240 (44.8%)

不正解(WA)の数: 237 (44.2%)

コンパイルエラー(CE)の数: 54 (10.1%)

実行時エラー(RTE)の数: 0

制限時間エラー(TLE)の数: 5 (0.9%)

であった。あらかじめ自分のPCで実行できることを確かめてから、コピー&ペーストで提出するので、CEは1割程度にとどまった。それを除くと、正答と誤答はほぼ同数ということになる。TLEのほとんどは無限ループに陥るものと推測される。ゼロ除算のようなRTEが検出されなかったのは意外であったが、エラー通知を取り逃して検出ができていない可能性もある。

次にSOJによる家庭学習の時間が増えているかどうかを調べる。そのため、日別時間帯別、曜日別に集計してみた。SOJの家庭学習の時間を増やす目的については、効果があった。

図4.1は、1年生がプログラムを提出した回数を日別に集計したものである。11/21(火)からの3週間は低調な滑り出しであった。学内と学外のネットワーク間にはファイアウォールが存在するため、学内からアクセスができないことも低調な要因の1つである。

12/14(火)の授業中に利用を促す周知をしたので、以降、提出件数が増えている。また、冬休みの期間中も提出件数が順調に伸びている。1/11(火)が最後の授業日で、それから定期試験日1/25(火)までは利用されない日が目立つ。

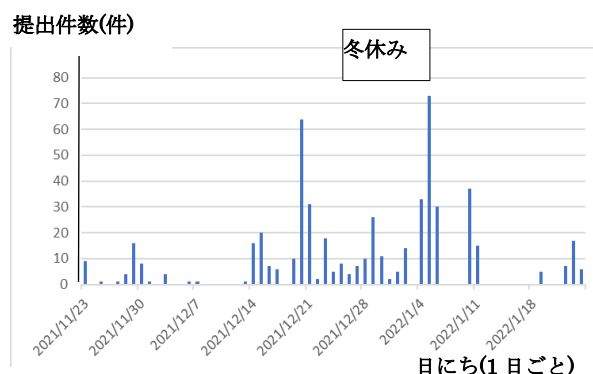


図 4.1 日ごとの提出件数

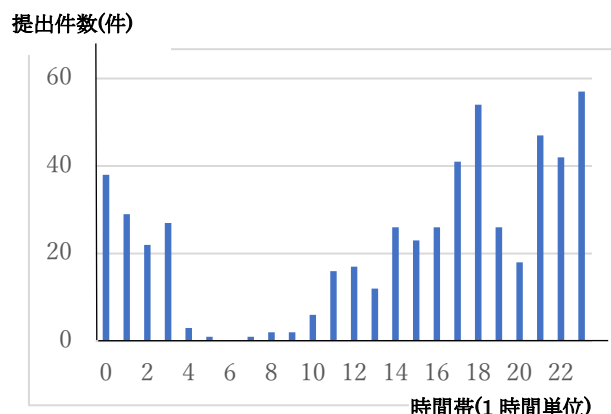


図 4.2 時間帯別の提出件数

次に、図 4.2 は、1 時間単位の時間帯別の提出回数を集計したものである。朝から午前中にかけての利用は少ない。夕方と深夜の時間帯に利用が多い。今回、深夜の時間帯に学習をしている人が予想外に多かった。19 時台と 20 時台の利用率が低いのは、バイトなどで忙しいためではないだろうか。SOJ は家庭学習を促進する一定の効果があつた。

図 4.3 は、曜日ごとの提出件数である。月～水曜日の利用が多い。プログラミング演習が火曜日開講であることと関係があるかもしれない。金曜日の利用が極端に少なく、土日の利用も少なかった。

今後は、週末に解答制限時間を設けたプログラミングコンテストを実施して、優秀者を称えることを企画したい。

提出件数(件)

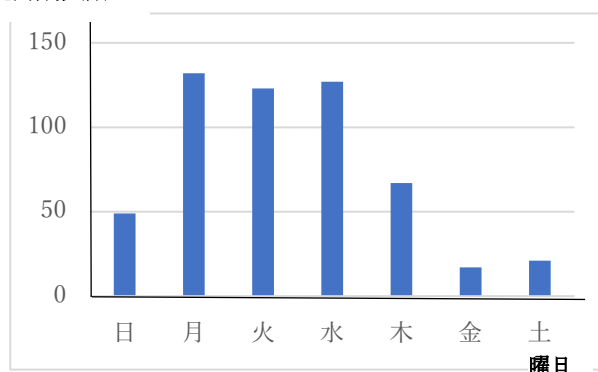


図 4.3 曜日ごとの提出件数

4.2 SOJ と実力試験の関係

プログラミング演習(国持クラス)では、第 1 回(受験者 20 名)と第 2 回(受験者 20 名)の実力試験を、第 14 回と第 15 回目の授業中にそれぞれ実施した。それぞれ 100 点満点で、プログラムの知識(満点 36)、読解(満点 20)、記述(満点 44)の 3 要素の能力を測定する。両テストの難易度は概ね等しくなるように作成したつもりである。2 つの実力試験の解答者数と平均点は以下の通りである。2 回のテストの間には、授業はなかったことから、読解と記述の能力の向上については、出題方式への慣れ、(SOJ を含む)家庭学習による実力の向上などが要因と考えられる。

表 4.1 実力試験の平均点の比較

	1 回目 (12/21)	2 回目 (1/11)
読解(満点 36)	5.0 点	13.0 点
知識(満点 20)	8.5 点	8.5 点
記述(満点 44)	16.1 点	26.8 点
全体(満点 100)	29.6 点	48.3 点

2 回実力試験の(能力測定尺度としての)妥当性を確かめるために、ここでは定期試験(1 年生全体で同一問題で実施)の得点との相関係数を算出してみた。第 1 回実力試験について相関係数は 0.71 であり、第 2 回実力試験について相関係数は 0.65 であった。やや強い相関が見られ、

2 回の実力試験は、能力測定の尺度として有効であると思われる。

次に、図 4.4 は 2 回の実力テストの得点と SOJ の AC 数の分布である。丸いマーカ(●)は 1 回目の、四角いマーカ(■)は 2 回の実力テストのそれぞれ得点を表している。AC の数と各実力テストの得点には強い相関はないように見える。それは、SOJ をほとんど利用しなかった人(図中の領域 A)、および利用はしたが得点に結びつかなかった人(領域 B)がいることが大きな要因である。とにかく、今回母集団が少なかったことから、特異なデータや外部要因によって相関係数が大きく変動したと考えられる。

得点(点)

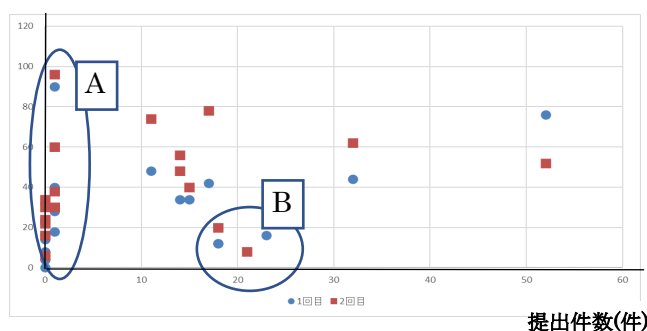


図 4.4 実力テストの得点と SOJ の AC 数の分布

その 1 つ目は、両実力テストの最高点の学生が、SOJ をほとんど利用していなかった。仮にその学生を除くと相関係数は+0.2 ほど増える。このように、優秀な学生が SOJ に興味を示さない、逆に SOJ を頻繁に利用した学生の成績が悪いというケースは十分にあり得る。また、コピーによる AC を不正に稼ぐ場合も対策が必要である。

2 つ目は、第 2 回の読解の問題のうち 1 つが、バブルソートのトレースを問うものであったが、他の授業で学んだ類題の答えをそのまま書き込んでしまっていた。したがって、第 2 回の読解は正しく能力を測ることができなかった。

表 4.2 実力テストの得点と AC の数の相関係数

	#1(12/21)	#2(1/11)
読解	0.52	0.13
知識	0.11	0.11
記述	0.37	0.31
全体	0.42	0.23

SOJ の結果と実力との相関は、以下の項目に考慮して、2022 年度に再度測定し直す予定がある。

- 不正行為への対策が必要である。他人の正解をコピー&ペーストして、提出する。
- 問題の質を上げ、学習範囲を網羅するのに十分な問

題量を用意する。

- 実力テストの測定精度を上げる。

5. あとがき

プログラミングの導入教育に活用するために、学外に専用サーをホスティングし、そこへ SOJ を移植し、稼働させた。そして、2021 年度 1 年後期必修科目「プログラミング演習」において試験運用した。SOJ の効果を測るために 2 回の実力テストを実施した。家庭学習時間においては寄与したが、SOJ の学習効果についてはデータが少ないため十分な検証ができなかった。授業評価アンケートの意見の中には「SOJ のおかげで力がついた」との意見が 1 件あった。

今後の取り組みについては、(1) CSS や Java Script を用いてフロントエンドの機能や可視化を充実させる。

(2) 対応するプログラミング言語の種類を増やすこと、不正コピー＆ペーストの検出と防止、採点過程の表示、計算時間や記憶容量の測定、などバックエンド側の機能を実現させる。

(3) プログラミング能力の実力診断、問題の自動選択など AI を用いた機能を付加する。

(4) 「コンピュータシステム実践演習 1」にて SOJ のコンテンツ作り、とくに問題作成を促進する。良問をたくさん作成することが SOJ の学習効果を上げる一番の課題である。

(5) 情報学部内部でプログラミングコンテストを実施して DP 到達度試験の 1 つの指標として、また副専攻の受講生に対する C 言語の副教材として、活用されるように今後企画していく。学習者向けの学習支援については、動画教材をパワーポイントで作成し、予復習に活用する。SOJ に掲載された問題を PDF として配布したい。

謝辞 本研究は「2021 年度 提案型教育研究 教育推進プロジェクト」の支援を受けた。関係各位に感謝の意を表します。

参考文献

- 1) 山下 颯也, 國持 良行. “プログラミング導入教育を目標にした SIST オンラインジャッジシステムの試作と提案,” 静岡理工科大学紀要, 29, pp.109-117 (2021-08-31)
- 2) 渡部有隆. “オンラインジャッジの開発と運用-Aizu Online Judge,” 情報処理, 56, 10, pp. 998-1005 (2015)
- 3) 松永賢次. “導入プログラミング教育におけるオンラインジャッジシステムの活用の試み,” 情報科学研究, 31, pp. 25-41 (2011)
- 4) 古谷勇樹, 林 真史, 山本 隆弘, 長尾 和彦. “RK-003 オンラインジャッジシステムと連携可能な Moodle プラグインの実装と比較 (K 分野: 教育工学・福祉工学・マルチメディア応用, 査読付き論文),” 情報科学技術フォーラム講

演論文集, 14,3, pp.89-94 (2015)

- 5) 長尾 和彦, 古谷 勇樹, 峯脇 さやか. “オンラインジャッジシステムのプログラミング演習への導入と評価,” 第 78 回全国大会講演論文集, 1, pp.537-538 (2016)
- 6) 古谷 勇樹, 林 真史, 山本 隆弘, 長尾 和彦, “オンラインジャッジシステムを用いたプログラミング学習環境の構築と比較,” 教育システム情報学会 2014 年度学生研究発表会
- 7) 松本彩花, 松原 南美, 渡邊遥輔, 多田 拓, 倉光 君郎, “Sumomo: ブロックチェーンを用いた教育用オンラインジャッジの提案,” 情報教育シンポジウム論文集 2019 (2019): 321-325.
- 8) 岩本舞, 中村真人, 小島俊輔, 中嶋卓雄, “不正コピー検出手法を備えたオンラインジャッジシステムの開発,” 情報処理学会論文誌教育とコンピュータ (TCE) 1.4 (2015): 38-47
- 9) 関根遼, 伊藤恵, 奥野拓. “数学文章題を利用したオンラインジャッジシステム向け問題自動生成手法の提案,” 教育システム情報学会 2019 年度第 5 回研究会, (2020).