

2025年3月17日（月）

数学ソフトウェアとフリードキュメント 37 @早稲田大学

# クラウド教育システム CES-Alpha によるコードの 自動採点：開発の経緯と授業への応用

東京女子大学/CES-Alpha 劉 雪峰

共同開発者：齋藤 裕(新潟大学) 田中 一成（早稲田大学）



## 自己紹介

- 劉 雪峰 (東京女子大学教授、兼業：(株) CES-Alpha代表取締役 )

2004年中国より来日、2009年東京大学で学位取得後、早稲田大学5年と新潟大学9年の勤務後、2023年に現職に。2022年8月株式会社CES-Alphaを設立。

**最終学歴：**2009年 博士 (数理科学) (東京大学) 取得

**経歴：**

2009年 IPA未踏プロジェクト代表開発者

2009年～2014年 早稲田大学 理工学術院 早稲田大学大石進一研究室で計算機援用証明について研究。

2014年～2023年 新潟大学

2023年～現在 東京女子大学

有限要素法の誤差評価理論・ナビエ・ストークス方程式の計算機援用証明の研究及び教育システムの開発などを行う。

## 話題提供 内容概要

- ◆ 授業実施を支援する**クラウド教育システム（CES-Alpha）**を開発
- ◆ [復習] 3 5 回では、「数学演習の自動採点」を紹介しました。
- ◆ 今回は、**コードの自動採点**の開発経緯、仕組みなどを説明
  - ⇒既存のコードの自動採点システム
  - ⇒CES-Alphaの自動採点の特徴
  - ⇒現在の対応言語
- ◆ **大学のプログラミング授業**での利用

# CES-Alphaの概要

## クラウド教育システム（CES-Alpha）

2015年に（当時）新潟大学の劉研究室による開発したオンライン教育システム。  
2022年8月株式会社CES-Alphaを設立。

CES-AlphaはWebブラウザを介して、主に以下の機能を提供している。

- 授業管理（授業HPの開設、レポート・アンケートの実施・集計等）
- 仮想プログラミング環境（動的な仮想マシン上でのJupyterLabの使用）
- オンラインサポート（学生・教員・TA間のチャット形式での質問対応）
- 数学演習問題の自動出題・自動採点
- **プログラミングコードの自動採点**

## 概要：CES-Alphaで実装したコードの自動採点

変数の値の入れ替え

変数xとyの値を入れ替えるために、以下の操作が必要です。

tmp = ア

x = イ

y = ウ

解答欄の「ア」「イ」「ウ」に答えを書いてください。

Your answer:

ア =

イ =

ウ =

解答提出

### 穴埋め形式

シンプルな穴埋め形式です。

スペースの有無などの入力の「ゆらぎ」にも対応しています。  
基本的なプログラミングの課題を自動採点できるため、プログラミング基礎科目に適しています。

素数判定

与えられた整数が素数であるかどうかを判定する関数 is\_prime() を作成してください。

コードのテンプレートをベースにしてコードを作成する:

```
def is_prime(n):  
    if n <= 1  
        return False  
    for i in range(2, int(n**0.5) + 1):  
        if n % i == 0:  
            return False  
    return True
```

解答提出

Ask AI for help

### コード提出式

テストコードの出力と一致するかどうかで正誤判定を行います。

初歩的な課題から、複雑な計算が必要になる課題まで対応できます。  
教員の判断で「Ask AI for help（AIに聞く）」ボタンを表示し、受講者にAIによる学習サポート機能を提供できます。

### その他

- AIによる学習サポート機能で受講者からの質問にも自動応答
- 教育向けに調整されたAIで、学生に与えるヒントのレベルが調整可能

# 既存のコード自動採点 システムの現状

# 既存サービス OR ソフトウェア

- Open Source: <https://github.com/DMOJ/online-judge>
- Aizu Online Judge: <https://judge.u-aizu.ac.jp/onlinejudge/>

会社による提供するサービス

- AtCoder (株): <https://info.atcoder.jp/> [コンテストの主催]
- 444 (株) : <https://techful.jp/> [スキルチェック, 教育機関向け?]
- paiza転職 <https://paiza.jp/> [スキルチェック]

(他に記載漏れのサービスがあれば、ご指摘ください。)

# 既存システムの考え方

## ➤使用対象者：

- スキルチェックやコンテスト形式で提供し、一定のプログラミング能力を持っていることを前提とする。
- 例えば、端末からの入力に対応できることが必須であり、一定の規則に従って結果を出力する必要がある。
- そのため、学生はプログラミングの入出力方法を学ぶまで、自動採点の演習を利用できない。

問題点 1：  
高校・大学のプログラミング授業の履修者に十分対応できていない。

## ➤自動採点方法・使用技術

- 与えられた入力に対し、出力された文字列を基に正誤判定を行う。

例：素数判定のコードでは、入力された値が素数であれば、Trueを出力する。そうではない場合、Falseを出力する。  
判定用のケースを事前に用意する。

例：Input: 18, Output: False; Input: 23, Output: True.

問題点 2：  
文字列を介して採点を行うため、期待される形式に変換しにくい場合、自動採点の対象とならない。

# 既存の採点方法で対応できない採点ニーズ

## 使用者の対応範囲

- プログラミング初心者の授業履修者でも利用しやすい形式で提供する。

## 採点方法

- 文字列の出力による正誤判定ではなく、変数の値を直接比較する方式を採用する。

## 丸め誤差への対応

- 数値計算の結果を採点する際には、一定の許容範囲を設けて正誤判定を行う。

## 問題作成のUI

- 教員が簡単に問題を作成できる仕組みを提供する。

# CES-ALPHAで実装した コードの自動採点

# 採点の方法

- Pythonのコードを例にして説明する。
  - $1 + 4 + 9 + \dots + n^2 > 100$  を満たす最小のnを求める。
- 以下の学生のコードをそのまま採点する。

```
sum = 0
n=0
while sum<=100:
    n=n+1
    sum=sum+n*n
```

特徴：  
学生のコードと正解のコードを同じソースコードで実行する

採点する際、学生のコードに赤の部分を自動的に追加して、コードを実行させる。(正解がn=7)

```
--Student's code --#
sum = 0
n=0
while sum<=100:
    n=n+1
    sum=sum+n*n
--Grading part --#
n_CORRECT = 7
if n == n_CORRECT:
    print("Result:OK")
    exit(0)
else:
    print("Result:NG")
    exit(1)
```

# 問題設定のページ（変数で採点する場合/Python）

問題プレビュー

while文の使用

閉じる

1 + 2 + 4 + ... + n<sup>2</sup> > 100 を満たす最小のnを求めるコードを作成してください。  
求めたnの値を変数nに格納してください。

コードテンプレートを使用してコードを作成する:

1 # ここでコードを書いてください。

2

コードをデバッグ

解答提出

## 採点方法の設定

プログラミング言語

Python 3

解答例

1 sum\_value = 0

2 n=0

3 while sum\_value<=100:

4   n=n+1

5   sum\_value=sum\_value+n\*n

6 |

解答例のコードは正しい値を返す。

入力引数(関数の作成を採点するとき)

各行に1つの答えを入力してください。関数の作成を採点しない場合、設定する必要はありません。

採点対象とする出力：出力変数（例：「x,y」）、または、入力を含めた関数（例：「myfunction(var1,var2)」）

n

採点の対象：この例では 採点変数はnとなります。

# 関数名で採点する場合

- 問題例

入力された数字a,bに対して、 $10*a+b$ の値を返す関数my\_sum(a,b)を作成する。

- 注意

- Pythonのコードでは、関数は再定義可能。
- Octave/MATLABなどの言語では、再定義した関数が構文のエラーとなります。
- よって、正解の関数の名前を自動的に修正するのは必要です。

採点用コード(赤の部分は自動生成)

```
--Student's code --#
def my_sum(a,b):
    return 10*a+b

--Grading part --#
#採点用のパラメタのケース
a=2; b=3;
#学生のコードの計算結果
result_student = my_sum(a,b)
def my_sum(a,b): #採点用のコード
    return 10*a+b
Result_CORRECT = my_sum(a,b)
if result_student == n_CORRECT:
    print("Result:OK")
    exit(0)
else:
    print("Result:NG")
    exit(1)
```

# 問題設定のページ（関数名で採点する場合[Python]）

問題プレビュー

関数の定義

閉じる

入力された数字a,bに対して、 $10 \times a + b$ の値を返す関数my\_sum(a,b)を作成する。

コードテンプレートを使用してコードを作成する:

```
1 def my_sum(a,b):
2     value = 0
3     #ここにコードを書いてください。
4
5     return value
6
```

コードをデバッグ

解答提出

採点方法の設定

プログラミング言語

Python 3

解答例

```
1 def my_sum(a,b):
2     value = 10*a+b
3     return value
4
```

入力引数(関数の作成を採点するとき)

```
3,5
2,7
1,5
```

採点対象とする出力：出力変数（例：「x,y」）、または、入力を含めた関数（例：「myfunction(var1,var2)」）

```
my_sum(var1,var2)
```

解答例のコードは正しい値を返す。

関数呼び出す時の引数

この例では採点対象は関数my\_sum(var1,var2) となります。

## 問題設定のページ（関数で採点する[MATLAB言語]）

### 採点方法の設定

プログラミング言語

MATLAB language (using Octave)

解答例

```
1 function min_value=my_min(numbers)
2   min_value = min(numbers);
3 end
```

入力引数(関数の作成を採点するとき)

```
[2,4,5]
[4,1,5]
[5,4,2]
```

採点対象とする出力：出力変数（例：「x,y」）、または、入力を含めた関数（例：「myfunction(var1,var2)」）

my\_min(var)

解答例のコードは採点のケースの入力に対して、正しい値を返してくれます。

採点ケースの入力リスト

採点の対象

注：Octave/MATLABなどの言語では、再定義した関数が構文のエラーとなりますので、採点用のコードでは、学生が作成した関数 **my\_min** と重複しないように、解答例の関数 **my\_min** の名前が自動的に **my\_min\_OK** に変更されます。

# CES-Alphaの採点方法の特徴

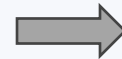
- 指定された変数で正誤判定を行う。
- 文字列への出力が不要となり、多くのタイプの変数の採点ができる。例えば、辞書タイプの変数の比較が可能。

例：

```
my_dict={'月曜日':1, '火曜日':2}
my_dict_OK={'火曜日':2, '月曜日':1}
my_dict == my_dict_OK # 結果が真
```

```
my_list=[1,2,3]
my_list_OK=[2,1,3]
my_list == my_list_OK # 結果が偽
```

```
my_set=set([1,2,3])
my_set_OK=set([2,1,3])
my_set == my_set_OK # 結果が真
```



注意：辞書の変数を文字列に出力する場合：

```
>>> print(my_dict)
{'月曜日': 1, '火曜日': 2}
>>> print(my_dict_OK)
{'火曜日': 2, '月曜日': 1}
```

文字列化にした結果が異なる。

## 数値計算結果の採点：許容誤差の対応

数値解析の授業では、反復法などの数値計算アルゴリズムで得た近似解は真の解とは誤差がありますので、一定の許容誤差の範囲で採点する必要があります。

CES-Alphaの採点では、数値タイプの変数に対して、指定された許容誤差で真偽判定を行なっています。

```
if norm(solution - solution_OK) < 1E-8)
    disp("Result:OK")
else
    disp("Result:NG")
```

MATLAB/Octave

## 現在の対応言語

**Python, MATLAB/Octave, R**

他の言語は要望に応じて追加できます。

注意：

- C言語などのスクリプト言語ではないプログラミング言語では、標準出力（端末への出力）の文字列を用いて採点する必要があります。

# コード採点における他の配慮点

コードの自動採点システムの設計では、以下の点に配慮しています。

## **セキュリティの問題**

悪意のあるコードの実行を防ぐため、**サンドボックス環境**を用意し、無限ループやファイルシステム・ネットワークへの不正アクセスを禁止しています。

## **計算資源の使用制限**

各プログラムの**実行時間**や**メモリ使用量**に制限を設け、計算資源の過剰消費やシステムのクラッシュを防いでいます。

## **待ち列の管理**

大量の提出が同時に発生した際にもスムーズに採点できるよう、優先度管理（待ち列管理）の仕組みを導入しています。

## 自動採点システムの実装

- Google Cloud Platform の環境(GAE) + 採点サーバー(Ubuntu)
- Google のクラウド環境はユーザーのアクセス数に応じて計算資源を調整してくれますので、授業で多数の学生がテストを受けることも大丈夫です。



Google App Engine  
(演習管理、演習実施など)



コード実行用の採点サーバー  
(タスク管理を使用する)

# デモ

- CES-Alphaのサイトで実演

<https://www.ces-alpha.org/>

# 問題設定：穴埋め式の対応

問題を定義

問題タイトル

変数の値のシフト

解答形式

テキスト入力 (穴埋めを含む)

問題文

変数xとyとzの値を  $x \rightarrow y, y \rightarrow z, z \rightarrow x$  にシフトするコード考えてください。

```
<pre>tmp = y;\ans{A} = \ans{B};\ans{C} = \ans{D};\ans{E} = \ans{F};</pre>
```

解答欄の「A」「B」「C」「D」に答えを書いてください。

解答

A=y  
B=x  
C=x  
D=z  
E=z  
F=tmp

複数の正解がある場合は、セミコロン「;」を使用して区切ってください。例：「PrimeNumber=3;13;23」。  
解答の文字列にセミコロンが含まれる場合は、「\;」を使用してください。

ヒント (任意)

変数に値を与えるために、関数の元々の値がなくならないように注意が必要です。



変数の値のシフト 閉じる

変数xとyとzの値を  $x \rightarrow y, y \rightarrow z, z \rightarrow x$  にシフトするコード考えてください。

```
tmp = y;  
A = B;  
C = D;  
E = F;
```

解答欄の「A」「B」「C」「D」に答えを書いてください。

ヒント：変数に値を与えるために、関数の元々の値がなくならないように注意が必要です。

回答：

A =

B =

C =

D =

E =

F =

解答提出

問題文の設定では、`\ans{A}`、`\ans{B}`、`\ans{C}`のような構文で、穴埋めの問題を設定できます。

# 穴埋め式問題の採点

特徴

- ・ 穴埋め形式で部分点を出せる
- ・ スペースなどの表記ブレに対応

####python

a, b = 1, 2

c =

(f"a+b={c}")

解答画面

解答の設定 :

ア=a+b;b+a

イ=print

## テキストの一致判定

模範解答 : a+b;b+a

a + b

a+b

3

b + a

OK

OK

NG

OK

模範解答 : print

print

print\_

ptint

p rint

OK

OK

NG

NG

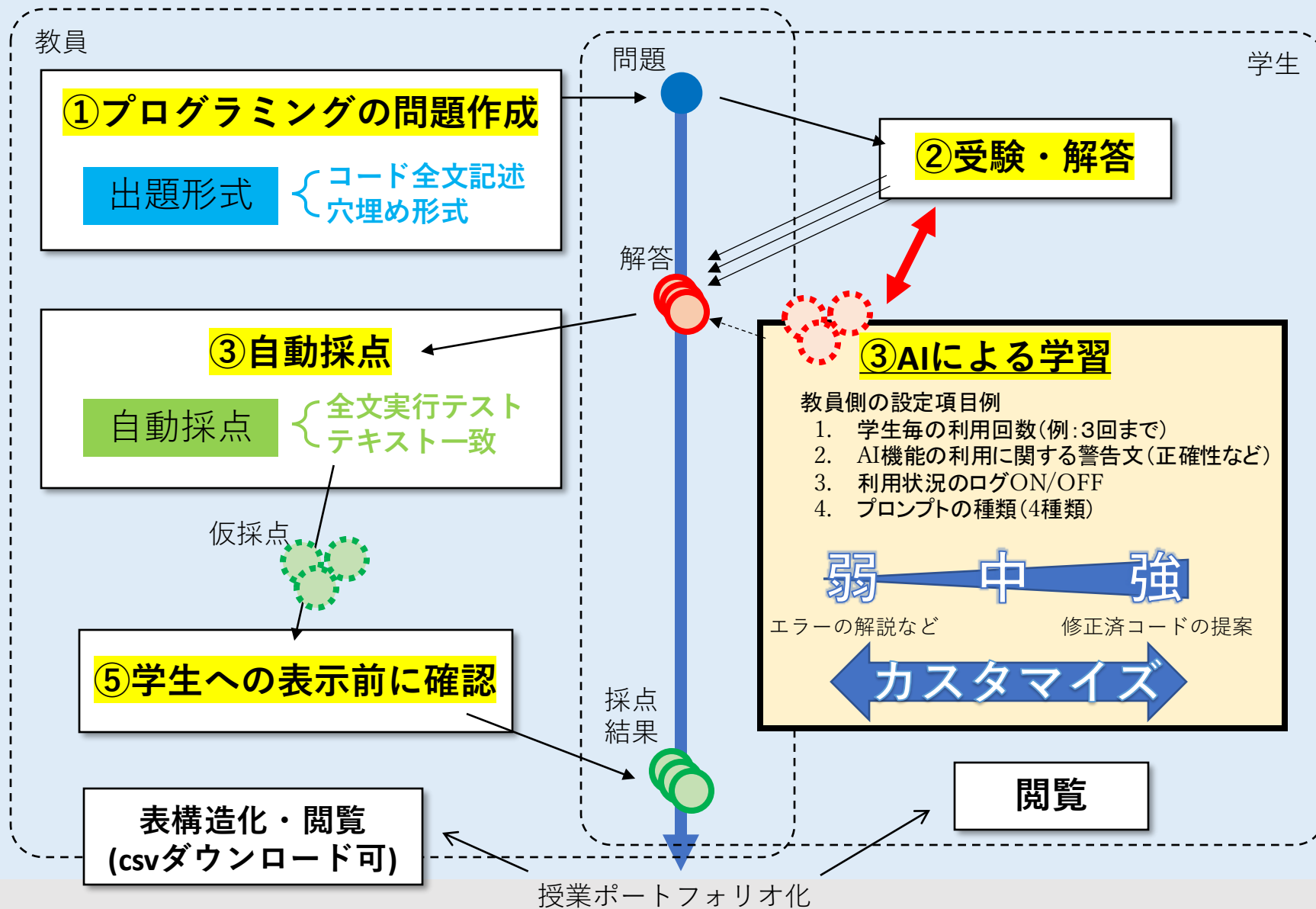
# 演習自動採点の使用イメージ

# プログラミング問題出題/自動採点

CES-Alpha

出題の流れ：  
自動採点・AIの適用タイミング

- 1 (教員) 問題作成
- 2 (学生) 受験・解答
- 3 (学生) AIによる学習
- 4 (内部) 自動採点
- 5 (教員) 自動採点結果の確認



# AIによる学習の使用イメージ

## AIカスタマイズ

ルール、入力形式、出力形式、例外処理など。  
プロンプティングのみのMy GPTsのイメージ。

システムの設定項目(例)

1. 学生毎の利用回数(例: **週30回まで**)
2. AI機能の利用に関する警告文(正確性など)
3. 利用のログ ON/OFF
4. **エラー文解説/関数解説など目的別プロンプト選択**

(エラー文から始まる学習イメージ)

入力

```
Cell In[22], line 1
print("Hello World!")
      ^
SyntaxError: incomplete input
```

次にpythonのエラー文を提示するので、  
エラーの理由を説明し、解決方法を提  
示してください。¥n 解答形式===¥n エ  
ラーの理由: ¥n 解決方法: ¥n===

プロンプト制御

エラーの理由:

このエラーメッセージは、Pythonで構文エラー (SyntaxError) が発生していることを示しています。具体的には、文字列が正しく閉じられていないためです。Pythonでは、文字列はシングルクォート (') またはダブルクォート (") で始まり、同じ記号で終わる必要があります。この場合、ダブルクォートで始まった文字列がダブルクォートで閉じられていないためエラーが発生しています。

解決方法:

この問題を解決するには、文字列を適切に閉じる必要があります。つまり、不足しているダブルクォート (") を行末に追加することで、文字列を正しく閉じます。

理由だけ  
指導して

システム制御

解決方法は不表示



AIによるプログラミング指導

AIに聞く  
コードのバグを分析します。

CES-AlphaのAI指導

# 学生による解答

各演習・小テストの出題リストに従った問題が配置され、  
**学生はコードを入力して解答**する。

入力したコードをオンライン実行できるので、**解答提出前のデバッグ機能**を提供している。

## (2) 最小値の計算

任意の3つの数に対してそれらの最小値を返す関数my\_min3を定義してください。  
ただし、関数minおよびmaxを使わないこと。定義した関数を呼び出す例も付けて確認してください。

コードテンプレートを使用してコードを作成する: [リセット](#)

```
1 def my_min3(a,b,c):  
2     #ここにコードを書いてください。  
3     min_value = a;  
4     if b<min_value: min_value =b  
5     if c<min_value: min_value =c  
6     return min_value  
7 print("実行の結果: ", my_min3(3,5,2))
```

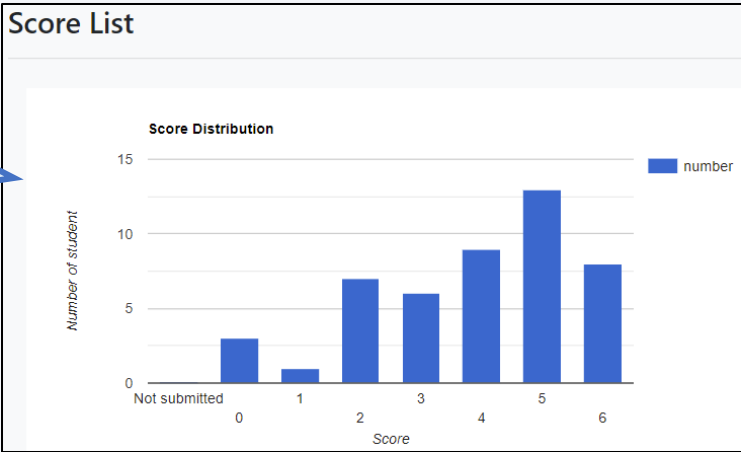
[コードをデバッグ](#)

実行の結果: 2

# 自動採点結果の表示

学生の解答は、自動採点後に下のような**分析結果**へと自動的にまとめられる。

得点の分布



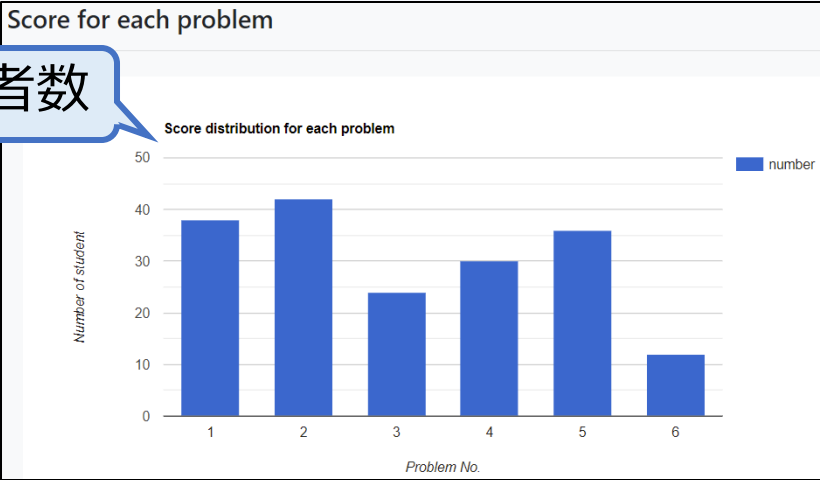
提出回数による  
ランキング

Top ranking of report submit

|   | Name | Score | Submit times |
|---|------|-------|--------------|
| 1 |      | 6     |              |
| 2 |      | 6     | 1            |
| 3 |      | 6     | 1            |
| 4 |      | 6     | 1            |
| 5 |      | 6     | 1            |
| 6 |      | 6     | 1            |
| 7 |      | 6     | 1            |
| 8 |      | 6     |              |

各履修者の得点の詳細

各問題の正答者数



Detail of score

|    | Name | Score 1 | Score 2 | Score 3 | Score 4 | Score 5 | Score 6 | Sum | Submit Number |
|----|------|---------|---------|---------|---------|---------|---------|-----|---------------|
| 1  |      | 1       | 1       | 0       | 1       | 1       | 1       | 5   | 1             |
| 2  |      | 0       | 1       | 0       | 1       | 1       | 0       | 3   | 1             |
| 3  |      | 1       | 1       | 0       | 1       | 0       | 0       | 3   | 1             |
| 4  |      | 1       | 1       | 1       | 1       | 1       | 1       | 6   | 1             |
| 5  |      | 1       | 1       | 0       | 1       | 1       | 1       | 5   | 1             |
| 6  |      | 1       | 1       | 0       | 1       | 1       | 0       | 4   | 1             |
| 7  |      | 1       | 1       | 0       | 0       | 0       | 0       | 2   | 1             |
| 8  |      | 1       | 1       | 0       | 0       | 0       | 0       | 2   | 1             |
| 9  |      | 1       | 1       | 1       | 1       | 1       | 0       | 5   | 1             |
| 10 |      | 0       | 1       | 1       | 1       | 1       | 0       | 4   | 1             |

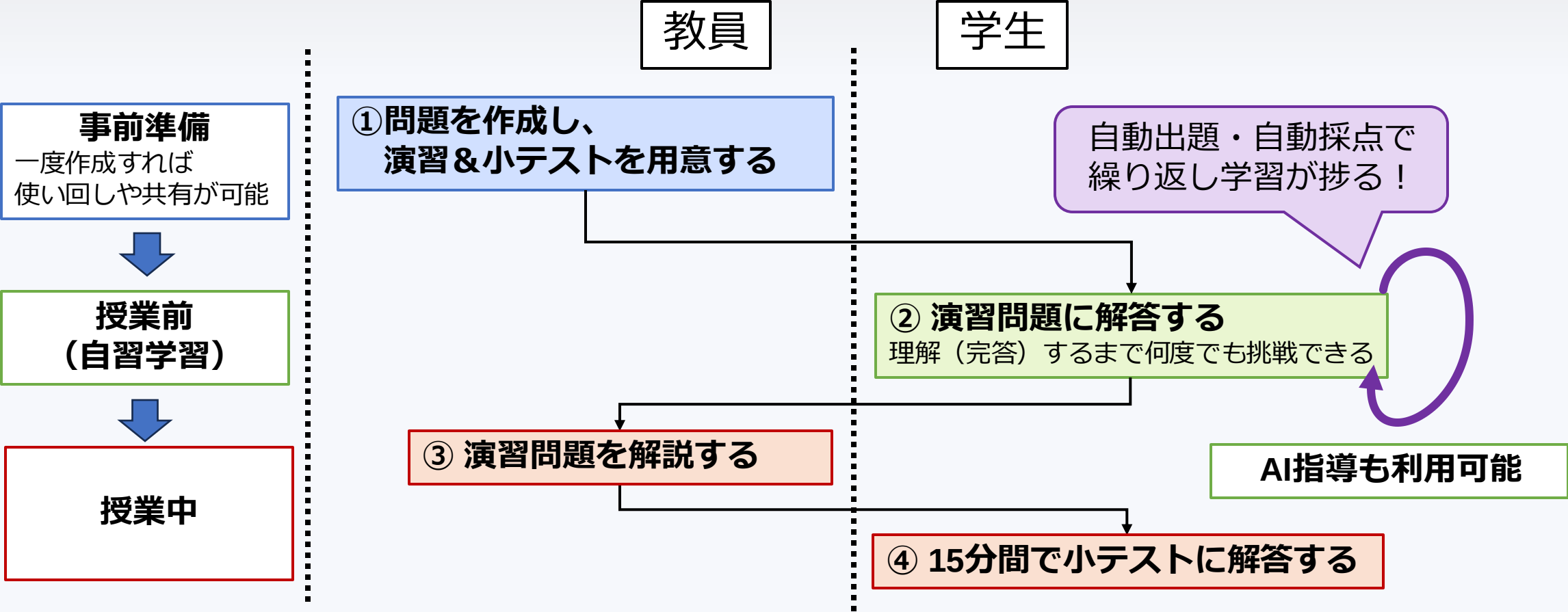
# 授業における使用状況

CES-Alpha のコード自動採点は、これまでに  
東京女子大学、東北大学、早稲田大学高等学院（高校）  
などの授業で使用されてきた。

- 東京女子大学: C, Python, Octave/MATLAB
  - 1年次演習 (Python)
  - 数値計算 (MATLAB)
  - 情報処理技法 (Cプログラミング)
  - シミュレーション AI / AII (MATLAB)
- 東北大学: Python言語 (機械学習の授業) 約450人の履修者
- 早稲田大学高等学院 (高校) : R言語

➤ 2024年11月、「先進的な数理・プログラミング教育を実現する次世代オンライン教育システム「CES-Alpha」」が、第21回日本e-Learning大賞で「プログラミング教育特別部門賞」を受賞！

# 東京女子大学での使用イメージ



## 学生アンケートの結果

（東京女子大学の学生向けの調査）

### 〈自動採点に関して〉

- 採点が早いため、**解答した内容を忘れないうちに問題を考え直す**ことができる
- すぐに採点結果を確認できる点。
- 自動採点により、**迅速なフィードバック**を得られる。

### 〈AI指導に関して〉

- **AI指導**があることで、授業内でわからなかったことを理解できたり、自分の考えたコード以外のコードを知ることが出来る。
- どこが間違っているのか**すぐに修正できて**ネットで調べるよりも早くできる
- 練習しやすかった。**AIが指導してくれる点**がとても良かった。
- AI指導機能で、自分の間違いや改善点を具体的に学べる。

# コードの自動採点システムの開発に関する展望

## 1. C言語などの言語への対応：

C言語などのスクリプト言語ではないプログラミング言語では、標準出力（端末への出力）の文字列を用いて採点する必要がある。

## 2. コードの品質評価

実行結果の正誤判定に加え、コード自身の効率性などをGPTを活用して評価する機能を追加する。

※ 現在、GPTによるコード評価は「AI指導」としてすでに導入済み。

## 3. 演習問題フォーマットの標準化

問題データのフォーマットを統一し、第三者間での問題共有を容易にする。

また、問題のオープンデータベースの構築も考えている。

## 4. 一部機能の無償提供

一定の範囲内で無料機能を提供し、使用者が自身で採点サーバーを準備する前提で利用可能とする。

# 謝辞

CES-Alphaの開発および一連の取り組みは、

- 新潟大学 2015年度授業改善プロジェクト
- 新潟大学 2021年度学長教育助成制度
- 科学技術振興機構事業 大学発新産業創出プログラム  
「みちのくアカデミア発スタートアップ準備資金」

のご支援を頂いております。