



<https://onlinejudge.u-aizu.ac.jp/>

チュートリアル

2020.10.18

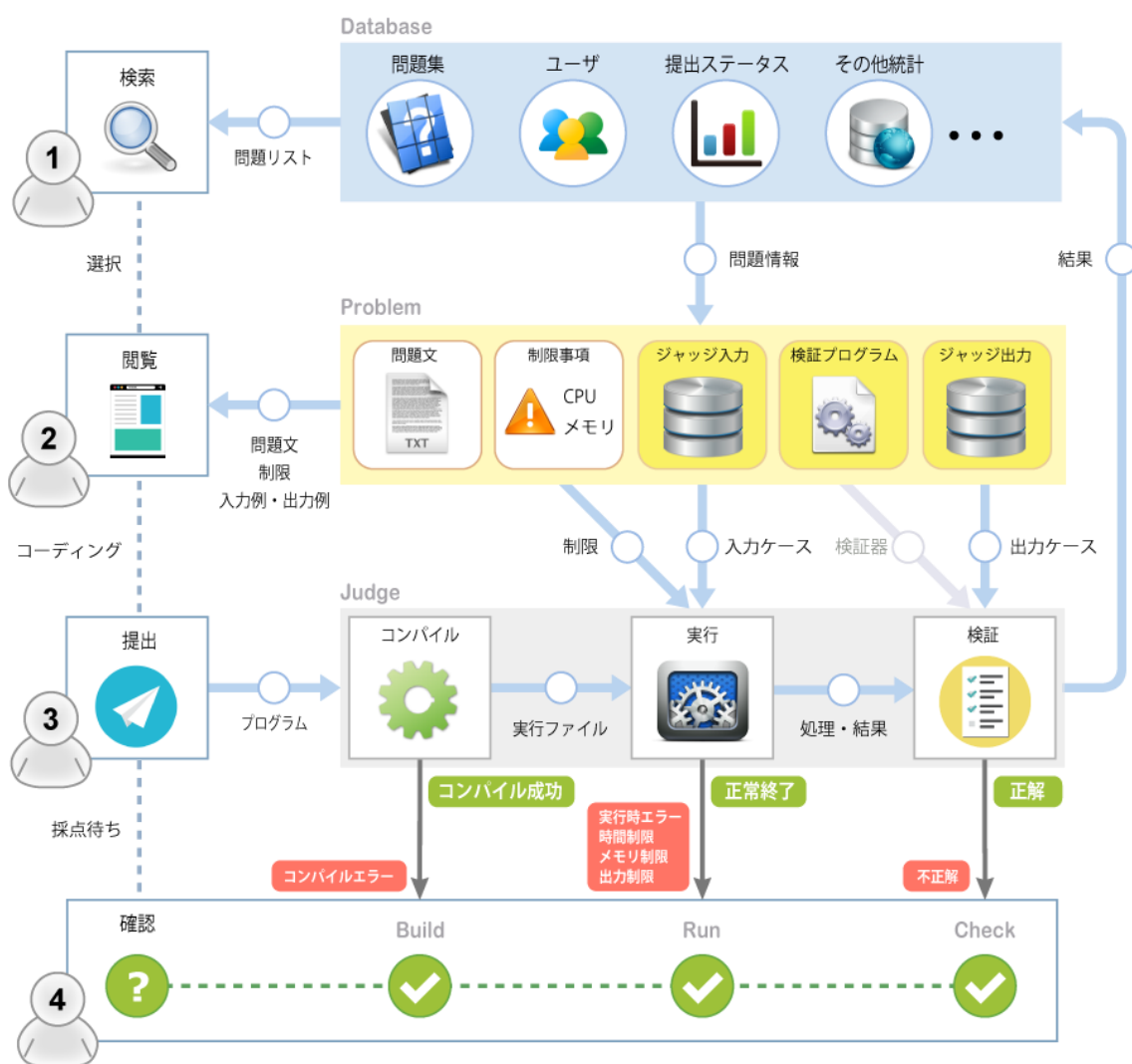
このチュートリアルでは、Aizu Online Judge (AOJ)の β バージョンの解説を行います。今後のリリースへ向けてユーザインタフェースや機能が変更される可能性がありますので、あらかじめご了承ください。引き続き旧バージョンのAOJも <http://judge.u-aizu.ac.jp> からご利用頂けます。

1. はじめに

AOJ (Aizu Online Judge) は無料で利用できるプログラミング問題のオンライン採点システムです。プログラミング初心者の学生からエンジニアまで、プログラミングスキルを磨きたい方、レベルアップを目指したい方に最適です。

1-1. オンラインジャッジシステムとは

オンラインジャッジとは、多くの演習問題にチャレンジすることができ、オンラインでコードを採点してくれるサービスです。各問題には、テストデータや検証器が準備されており、提出されたコードの正誤とその効率が即座に判定されます。一般的に、オンラインジャッジシステムは下図のようなしくみになっています。



オンラインジャッジを通して、利用者は以下のことを繰り返し追求することで、プログラマに必要な資質・能力と知識を体得することができます。

- 文書から要件を正確に理解する。
- 計算効率・メモリ使用量を的確に見積もり、コンピュータの資源を考慮したアルゴリズムを設計する。
- 設計通りのアルゴリズム、あるいは頭の中で描いた手順を、バグのないプログラムとして実装する。
- プログラマに必要な、データ構造とアルゴリズムの幅広い知識を獲得する。

オンラインジャッジを使ったプログラミングの魅力は、ゲーム感覚で楽しく学習が進められることです。問題解決とコーディング、正解による成功体験を繰り返すことで、知識・技術を身に付けることができます。

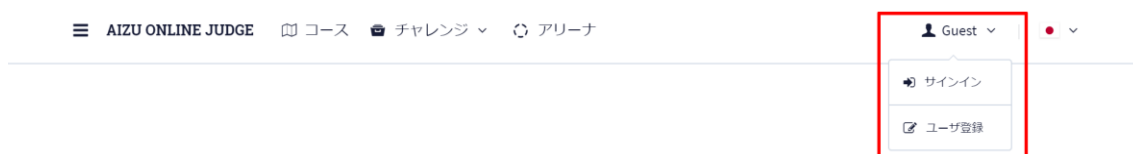
1-2. Aizu Online Judge

Aizu Online Judge (AOJ) は会津大学が開発・運用するオンラインジャッジシステムです。プログラミング初学者向けのコース、ICPC, JOI, パソコン甲子園、有志コンテスト等の問題を 2000 問以上収録しています。入門レベルの問題も多く含まれるため、プログラミング、アルゴリズムとデータ構造など、学校の授業の教育ツールとしても活用することができます。

2. ユーザ登録

2-1. サインアップ

プログラムを提出するためには、ユーザ登録を行う必要があります。AOJ へのユーザ登録は以下のようにヘッダーメニュー右側の「Guest」→「ユーザ登録」から行います。



新規ユーザ登録ページは以下のような構成になっています。

ユーザ登録をする前に...

利用規約に関する説明と注意書きをお読み下さい：[利用規約について](#)

アカウントID*:	<input type="text"/>	① 半角アルファベット、半角数字、半角記号
パスワード*:	<input type="password"/>	4文字以上 (Do not use special characters like \ and ')
パスワード(確認)*:	<input type="password"/>	

名前*:	<input type="text"/>	② 半角アルファベット、半角数字、半角記号、空白
アバター:	<input type="button" value="Choose File"/> No file chosen	③
所属:	<input type="text"/>	④ 半角アルファベット、半角数字、半角記号、空白
在住または出身国*:	<input type="text" value="Japan"/>	
生まれた年:	<input type="text"/>	
メールアドレス:	<input type="text"/>	
URL:	<input type="text"/>	あなたのホームページやBLOG

登録を行う前に、「利用規約について」のページにて注意事項を確認しましょう。主にプログラムの提出に関する注意事項について記載されています。登録に必要な主な情報は以下の通りです。

- ① **アカウント ID** : サインインに用いる ID。サイト内で使われるユニークなユーザ ID となります。
- ② **名前** : ユーザステータスやランキングの表示に用いられる氏名 (ニックネーム)。日本語の全角文字は使用できませんので注意してください。
- ③ **アバター** : ユーザステータスやランキングの一部に表示されるアイコン画像。この項目はオプションですので未設定でも問題ありません。
- ④ **所属** : 学校や所属機関の名称など。こちらも全角文字は使えませんのでアルファベット文字で設定します。
- ⑤ **表示言語** : サイト内で用いる言語 (Japanese または English)。サイトのメニューや問題文 (存在する場合) が設定された言語で表示されます。
- ⑥ **プログラミング言語** : 普段使用するプログラミング言語。プログラムを提出する際にプログラミング言語を選択することができますが、ユーザ登録で指定した言語がデフォルトとして設定されます。
- ⑦ **提出コードの公開** : public 設定の場合、提出されたソースコードが他のユーザに公開されます。private 設定の場合は、当該ユーザのみが自分のソースコードにアクセスすることができます。ソースコードを公開することで他のユーザの学習に貢献できるだけでなく、レビューされることで自分のコードの改善にも繋がりますので、public 設定にすることをお勧めしています。

入力が終了したら送信します。正しい入力が行われ「ようこそ AOJ へ！」というメッセージが表示されれば登録完了となります。

2-2. サインイン

オンラインジャッジの機能をフルに活用するために、サインインを行います。ヘッダーメニューの「Guest」→「サインイン」からダイアログを開き、登録したユーザ ID とパスワードを入力し、サインイン（ログイン）します。サインイン後は、プログラムを提出できるだけでなく、以下の機能面で便利になります：

- 検索画面等で、自分が既に解いている問題のチェックマーク及び進捗・得点の確認
- 自分が提出したソースコードの閲覧
- 問題にブックマークを付けることが可能
- その他、データを更新するための機能が有効（リコメンド投票やタグ付けなど）

2-3. 登録情報の更新

サインイン後は、ユーザメニューにて登録情報とパスワードの更新を行うことができます。

3. メニュー

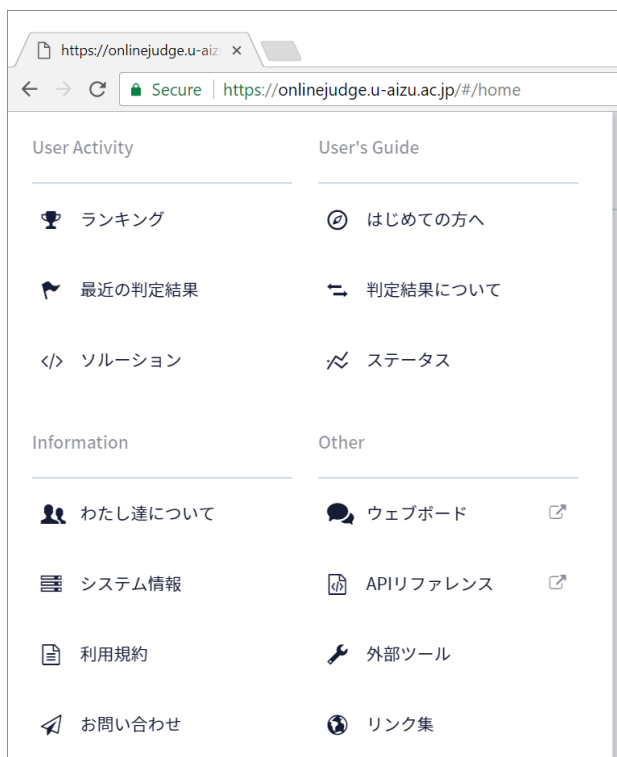
AOJ のヘッダは次のような構成になっています。



AOJ は主に、コース、チャレンジ、アリーナの3つの項目から構成されています。これらは以下の目的ごとに、それぞれ特化した検索機能やユーザインタフェースを提供します。

- コースでプログラミングの基礎力をつける
- コンテストなどの過去問に挑戦する
- コンテスト形式や演習形式で問題セットにチャレンジする

ヘッダの左端のアイコンから次のようなメニューを開くことができます。



メニューは [User Activity] 、 [User's Guide] 、 [Information] 、 [Other] の4つのセクションから構成されています。セクション内の各項目は、対応するページのリンクです。

[User Activity] は日々更新されるユーザの活動状況や統計情報を提供します。AOJ では、正解した問題数など様々な基準でユーザがランキングされます。また、ユーザの最近の提出状況などをリアルタイムで観察することができます。

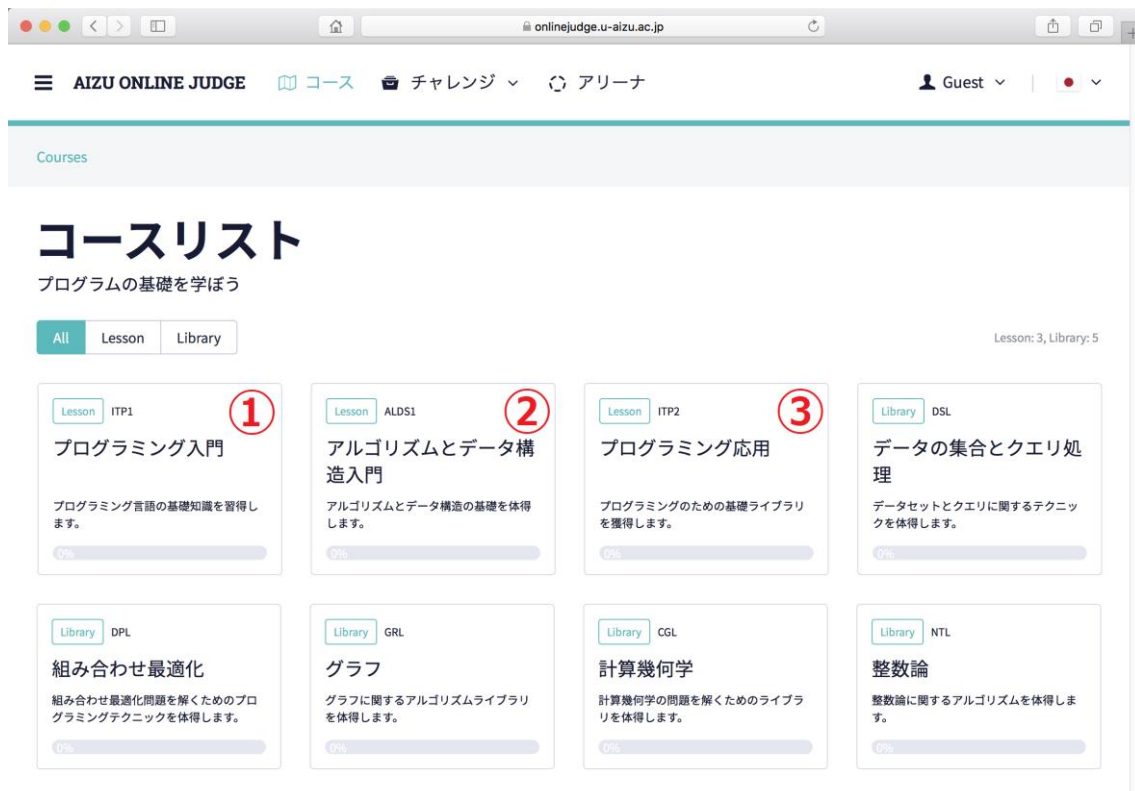
[User's Guide] には、主にはじめて AOJ をご利用される方向けの情報がまとめられています。例えば、ジャッジの判定結果の意味や、統計情報の内容などを確認することができます。

[Information] 、 [Other] には、システムやコンパイラの詳細、内部情報、開発者向けの API など、より興味に特化した詳細情報が提供されています。

4. 問題の検索

4-1. コース問題の検索

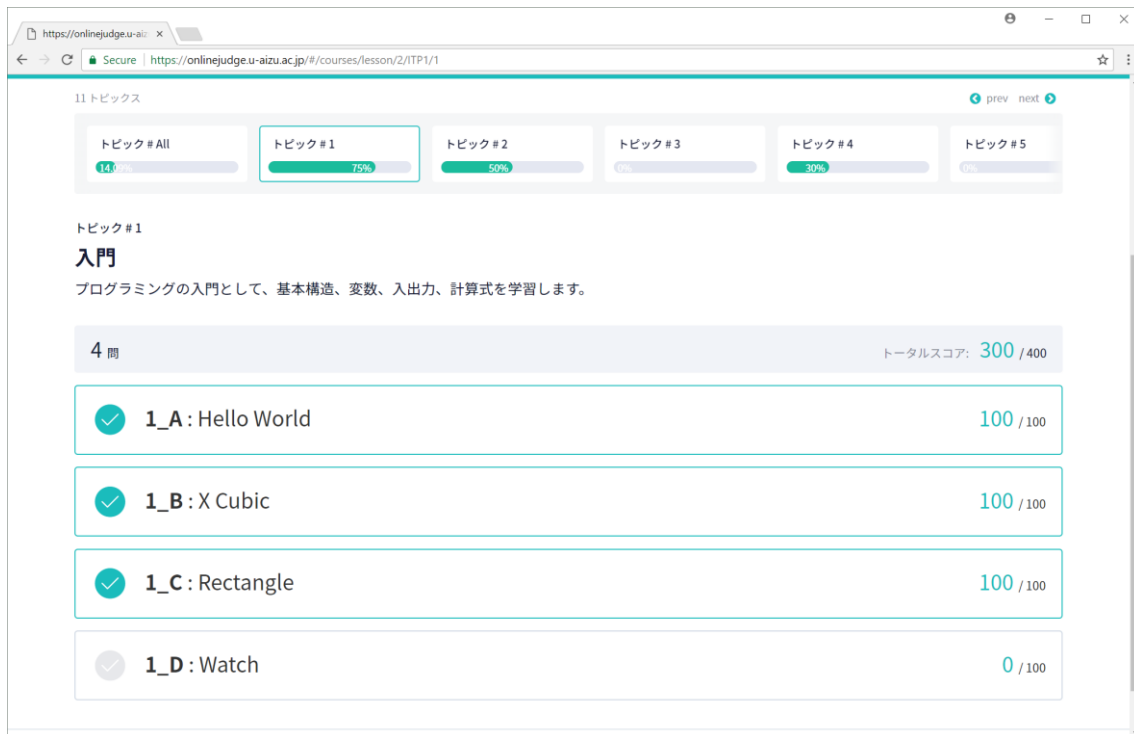
問題は主にコースとチャレンジに分類されています。それぞれに特化したユーザインタフェースから問題を選択することができます。（ここではコースについてのみ解説します。）**コース**には、プログラミングやアルゴリズムとデータ構造の入門のための問題セットが収録されています。主に各種プログラミング言語の文法や典型的な知識を養うことを想定した問題セットとなっています。トップページは以下のようにいくつかのコースに分類されており、それぞれカードで表されています。カードにはコースのタイトルと概要に加えて、サインイン時には進捗を確認することができます。



- ① **プログラミング入門**は、基本制御構造などの文法を確認するための基礎的な問題で構成されており、これから新たにプログラミング言語を覚えたい方にお勧めします。
- ② **アルゴリズムとデータ構造入門**は、プログラマとして必要なアルゴリズムとデータ構造の基礎的な問題から構成されています。何かしらのプログラミング言語の文法の基礎を一通り習得し、さらにプログラマとしての基礎技術を身に付けたい方にお勧めします。
- ③ **プログラミング応用**は、アルゴリズムとデータ構造の基礎を理解した上で、標準的なライブラリの使い方や汎用的なプログラミングテクニックを養うための問題です。

その他のコースは、古典的なアルゴリズムやデータ構造の知識を養うための問題です。これらの問題の中には高度なテクニックを要する難問も含まれますので、無理に進める必要はありません。

カードを選択することで以下のようなコースのページを展開します。



各コースはいくつかのトピックから構成され、ページ上部のカードから各トピックの達成度 (%) を確認することができます。さらにトピックはいくつかの問題で構成され、カードを選択することで中の問題リストを展開します。

問題リストでは、トピック及び各問題の得点を確認することができます。問題のカード (リンク) をクリックすると問題ページが開きます。

5. 問題の解答とプログラムの提出

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の本質的な活動について解説します。問題を解いて、プログラムを提出してみましょう。

5-1. 問題ページの構成

ファインダーや検索ページで問題カード（リンク）を選択すると、以下のような問題ページが開きます。

The screenshot shows the AIZU ONLINE JUDGE interface for a problem titled 'PCK - Prelim 0358'. The page is annotated with red boxes and numbers 1-1 through 1-5:

- 1-1**: A red box highlights the problem constraints: "Time Limit: 1 sec, Memory Limit: 262144 KB, isSolved: [dropdown]".
- 1-2**: A red box highlights the problem title "買い物" and the introductory text describing the scenario of buying a book from a friend.
- 1-3**: A red box highlights the "Input" section, which specifies the input format "m f b" and provides a sample input "1000 3000 3000".
- 1-4**: A red box highlights the "Output" section, which explains that the output should be the minimum amount to borrow, or "NA" if the user has enough money. A sample output "2000" is shown.
- 1-5**: A red box highlights the "Sample Input 2" and "Sample Output 2" section, showing a second example with input "5000 3000 4500" and output "0".

The screenshot shows a typical AOJ problem page layout. It is divided into five main sections, each highlighted with a red box and a circled number:

- 2-1:** Language selection dropdown (C++).
- 2-2:** Code editor area containing C++ code for a simple comparison problem.
- 2-3:** Submission controls including 'Clear' and 'Submit' buttons.
- 3:** Judge status overview, including a progress bar with steps (Submitted, Sent to Judge, Build, Run, Resource Limit Check, Result Check, Presentation Check) and a table with columns for Lang, Status, Judge, Time, Memory, and Code.
- 4:** Test case overview table with columns for Case #, Status, Time, Memory, In, Out, and caseName.
- 5:** Footer area containing navigation links (Web Board, Solutions, Statistics, Tagging) and source information (15th PC Koshien, Preliminary Round, created at: 1/1/2018, 3:24:53 AM).

問題ページは主に以下のパーツから構成されています。

- ① **問題文**： 問題の内容が書かれています。
- ② **コーディングフォーム**： コーディングを行い（または作成したコードを張り付け）、プログラムをジャッジに提出するエリアです。
- ③ **ステータスビュー**： プログラムを提出した直後に、ジャッジからの返答を確認するビューです。
- ④ **テストケースビュー**： ジャッジが完了した後に、各テストケースに対する判定結果とジャッジデータを確認するビューです。
- ⑤ **フッタ**： 解答リストや統計など、当該問題に関する補助的な情報へのリンクがまとめられています。

3-2. 問題文の構成

各問題は、「課題について、与えられた**入力**に対して正しい**出力**を行うプログラムを作成せよ」という出題形式になっています。具体的には、問題文は主に以下の項目から構成されています。

- ①-1 **ヘッダ**：主に問題に関する制限事項が書かれています。
Time Limit、Memory Limit はそれぞれ、問題を解答するために利用できる CPU の時間とメモリ量です。サーバー側（ジャッジ側）で提出プログラムの実行時間やメモリ使用量がこれらの値を超えてしまうと不正解となります。アルゴリズムを設計する際の参考にします。
- ①-2 **問題文**：問題の内容が書かれている本文です。背景や物語を交えながら、プログラムが満たすべき仕様が明確に記述されています。
- ①-3 **入力(Input)**：問題の入力に関する説明が書かれています。プログラムはここで定義されている入力形式で入力データを読み込まなければなりません。特に指定がない限り「標準入力」から読み込みます。
- ①-4 **出力(Output)**：問題の出力に関する説明が書かれています。プログラムはここで定義されている形式で出力を行わなければなりません。特に指定がない限り「標準出力」へ出力します。プロンプトメッセージなど余分な文字列を出力してしまうと不正解となるので注意しましょう。
- ①-5 **入出力例(Sample Input X, Sample Output X)**：入力例は、ジャッジデータとして与えられる入力の例であり、入出力の形式や考えられるパターンをより具体的に確認するためのものです。出力例は入力例に対する正しい出力を示します。※出力例と一致したプログラムが正解になるとは限らないことに留意してください。提出されたプログラムはより厳格でサイズが大きいデータを用いて判定されます。

3-3. プログラムの提出

「②コーディングフォーム」は、解答コードをジャッジシステムに提出するためのフォームで、主に以下のパーツから構成されています。

- ②-1 **プログラミング言語選択**：提出するプログラムの言語を選択します。デフォルトであらかじめ登録された言語が設定されます。現在 AOJ では以下のプログラミング言語に対応しています。

C, C++, C++11, C++14, C++17, JAVA, C#, Python, Python3, Scala, Go, Rust, JavaScript, Ruby, PHP, D, OCaml, Haskell, Kotlin

- ②-2 **エディタ**：解答プログラムを記述するテキストエリアです。プログラムを作成したツールなどからソースコードをコピー&ペーストして貼り付けます。テキストエリアは簡易的なエディタになっていますので、ブラウザ上から直接コーディングを行うこともできます。（※上図では、例としてあらかじめコードが埋め込まれています）
- ②-3 **Submit (提出) ボタン**：ジャッジに解答プログラムを送信します。

3-4. 結果の確認

提出ボタンを押すと「③ステータスビュー」と「④テストケースビュー」が以下のように更新されていきます。

JUDGE ID: 2697729 By YUTAKA

Submitted Sent to Judge Build Run Resource Limit Check Result Check Presentation Check

Lang Status Judge Time Memory Code

C++ CE 0/16 00:00 s 0 KB 16 B

rep/code.cpp: In function 'int main()': rep/code.cpp:1:11: error: expected ']' at end of input int main(){ ^

TEST CASE

Case #	Status	Time	Memory	In	Out	caseName
> # 1	CE	00:00 s	0 KB	15 B	5 B	00_sample_01.in

ステータスビュー右上 (③-1) にジャッジの進捗状況と結果が表示されます。ジャッジの進捗は「WAITING JUDGE (待ち)」→「JUDGING (判定中)」→「結果」と移り変わります。上図の最初の例では、試しに文法にミスがあるコードを提出したので結果が「COMPILE ERROR (コンパイルエラー)」となっています。

ステータスビュー中央 (③-2) の直線上のチェックマークはジャッジのステージを表し、ステージのどこまで成功したかを確認するための補助的なビューです。ステージは左から右へ Submitted → 進捗が「WAITING JUDGE」に変化 → Sent to Judge → 進捗が「JUDGING」に変化、と遷移していき、以降のステージも左から右に順番にチェックされ、失敗したところで結果が決定するか（不正解）、あるいはすべてのステージをパスし結果が「ACCEPTED（正解）」となります。

コンパイルエラーを修正して、再度提出してみましょう。今度は、以下のような結果が表示されました。

JUDGE ID : 2693350 By YUTAKA WRONG ANSWER

Submitted Sent to Judge Build Run Resource Limit Check **Result Check** Presentation Check

Lang	Status	Judge	Time	Memory	Code
C++	WA	2/16	00:00 s	3112 KB	205 B

③-3

④

TEST CASE

Case #	Status	Time	Memory	In	Out	caseName
> # 1	AC	00:00 s	3112 KB	15 B	5 B	00_sample_01.in
> # 2	AC	00:00 s	3088 KB	15 B	2 B	00_sample_02.in
∨ # 3	WA	00:00 s	3016 KB	14 B	3 B	00_sample_03.in

input: 500 1000 2000 output: NA

今回は、間違ったコードを提出したので結果が「WRONG ANSWER」となりました。前回の COMPILE ERROR から少し進歩し、Build、Run、Resource Limit Check をクリアしています。コンパイルと実行が成功している場合は、ステータスビュー下部 (③-3) で、ジャッジ結果に加えてプログラムが使用した CPU 時間とメモリを確認することができます。

ジャッジの状態と判定結果をさらに詳しく確認します。 ジャッジは以下の手順で進行し判定を行います：

ステージ	進捗／結果	提出されたプログラムの状態・詳細
Submitted	WAITING JUDGE	サーバ側の API にプログラムが届き、キューに追加されました。 順番待ちの状態です。
Send to Judge	JUDGING	キューから取り出され、ジャッジに送信されました。 判定結果を待っている状態です。
Build	CE COMPILE ERROR	コンパイルに失敗しました。 コンパイルエラーメッセージを確認してください。
Run	RE RUNTIME ERROR	実行中にエラーが発生しました。 不正なメモリアクセス等による Segmentation Fault、スタックオーバーフロー、ゼロによる割り算など多くの原因が考えられます。
Resource Limit Check	TLE TIME LIMIT EXCEEDED	問題に指定された制限時間内にプログラムが終了しませんでした。 各テストケースについて、ジャッジは問題の制限時間の + 1 秒程度の間プログラムを実行し続けます。 ただし、状況によって強制終了される場合があります。
	MLE MEMORY LIMIT EXCEEDED	問題に指定された以上のメモリを使用しました。
	OLE OUTPUT LIMIT EXCEEDED	制限を越えたサイズの出力を行いました。
Result Check	WA WRONG ANSWER	審判データと異なる出力データを生成しました。 特別なジャッジについては、問題専用に準備された検証器の結果や精度を考慮します。
Presentation Check	PE PRESENTATION ERROR	ほぼ受理される出力を行っていますが、出力形式に誤りがあります。 余計な空白や改行を行っていたり、あるいは必要な空白や改行を出力していません。
	AC ACCEPTED	上記すべての審査において拒否されなかったため、"受理"されました。

ジャッジ結果の詳細を確認するために、④テストケースビューを見てみましょう。 オンラインジャッジでは、複数のジャッジデータファイル（テストケース）を用いて正誤・性能判定が行われます。 テストケースビューでは、各テストケースの判定結果と使用したリソースを確認することができます。 AOJ では、あるテストケースについて不正解だった場合、それ以降のジャッジを中断し即座に結果を返します。

今回の例では、最初の2つのテストケースに対してはACしていますが、3つ目のテストケースで WRONG ANSWER となっています。テストケースビューでは、各ケースの左端のアイコンをクリックすることで、ジャッジデータとして利用されたテストケースの内容を確認することができます。例では、WRONG ANSWER になった3つ目のテストケースを展開しています。左側に input、右側に output (正しい出力) が表示されます。ジャッジデータはデバグの参考にもなりますので、必要に応じて活用しましょう。

ジャッジ結果にかかわらず何度でも再提出を行うことができます。正解するまで、あるいは納得のいくコードや性能が得られるまで、何度でもチャレンジしましょう。

バグを修正して、再度プログラムを提出してみます。以下のようにすべてのステージにチェックが付き、結果が ACCEPTED になれば正解です。

JUDGE ID: 2693351 By YUTAKA
ACCEPTED

Lang	Status	Judge	Time	Memory	Code
C++	AC	16/16	00:00 s	3092 KB	205 B

TEST CASE

Case #	Status	Time	Memory	In	Out	caseName
> # 1	AC	00:00 s	2984 KB	15 B	5 B	00_sample_01.in
> # 2	AC	00:00 s	3032 KB	15 B	2 B	00_sample_02.in
> # 3	AC	00:00 s	3000 KB	14 B	3 B	00_sample_03.in
> # 4	AC	00:00 s	3092 KB	12 B	4 B	10_simple_01.in
> # 5	AC	00:00 s	2988 KB	16 B	5 B	10_simple_02.in

3-5. ジャッジキューの閲覧

一般的なオンラインジャッジシステムでは、他のユーザの提出履歴を閲覧することができます。AOJではジャッジキューの状態をオープンにするため、最近の提出状況やユーザの提出履歴を公開しています。メニューの「最近の判定結果」から、以下のようにユーザの最近の提出状況を確認することができます。

Run #	Status	userID	ProblemID	Problem	Judge	Lang	Time	Memory	Code	Submission Date
2696619	AC	Ryoga_0212	0097	Sum of Integers II	4/4	C++	00:00 s	2788 KB	316 B	2/1/2018, 7:35:27 P M
2696618	AC	macha2	0583	Common Divisors	5/5	C++	00:00 s	3168 KB	523 B	2/1/2018, 7:35:22 P M
2696617	AC	kyuna	0104	Magical Tiles	1/1	Python3	00:02 s	5624 KB	271 B	2/1/2018, 7:34:58 P M
2696616	WA	Ryoga_0212	0097	Sum of Integers II	0/4	C++	00:00 s	2844 KB	369 B	2/1/2018, 7:34:44 P M
2696615	AC	kyuna	0104	Magical Tiles	1/1	Python3	00:03 s	5628 KB	271 B	2/1/2018, 7:33:32 P M
2696614	AC	kyuna	0104	Magical Tiles	1/1	Python3	00:03 s	5632 KB	271 B	2/1/2018, 7:33:31 P M
2696613	AC	kyuna	0104	Magical Tiles	1/1	Python3	00:03 s	5628 KB	271 B	2/1/2018, 7:33:27 P M
2696612	AC	herbager000	2824	Coastline	2/2	C++14	00:35 s	3496 KB	568 B	2/1/2018, 7:32:59 P M
2696611	AC	morii	ALDS1_2_A	Bubble Sort	10/10	Python3	00:03 s	5604 KB	343 B	2/1/2018, 7:31:27 P M
2696610	TLE	aquef	ALDS1_11_D	Connected Components	20/25	C++11	01:99 s	3224 KB	1293 B	2/1/2018, 7:30:43 P M
2696609	TLE	cherry_57	ITP1_11_D	Dice IV	3/20	Python3	09:99 s	8240 KB	1728 B	2/1/2018, 7:30:00 P M
2696608	AC	J01339	ITP1_3_B	Print Test Cases	1/1	C++	00:00 s	3076 KB	418 B	2/1/2018, 7:29:13 P M
2696607	MLE	vjudge2	ALDS1_3_C	Doubly Linked List	9/10	JAVA	04:30 s	703884 KB	1758 B	2/1/2018, 7:27:10 P M

3-6. 問題を解き終わったあと

オンラインジャッジでは、問題を解き終わった後、または問題につまずいてしまったときなど、学習の参考になる情報を公開しています。問題ページの下部の⑤フッタには、当該問題に関連する以下の情報へのリンクがまとめられています。

- **Web Board (掲示板)** : 問題に関する議論やアナウンスが行われている掲示板へ移動します。
- **Solutions (解答例)** : 解答例のページへ移動します。問題に正解した他のユーザのソースコードを閲覧することができます。
- **Statistics (統計)** : 問題に対する統計情報ページへ移動します。言語ごとの正解数やステータスの各種統計情報を閲覧することができます。
- **Tagging (タグ)** : 問題に関連付けられたタグ・カテゴリリストのページへ移動します。カテゴリとタグを閲覧・追加することができます。

4. オンラインジャッジシステムの機能の活用

プログラミングの学習を効率的に、楽しく進めるための補助機能をいくつかご紹介します。モチベーションを維持しながら、継続的に問題を解いていきましょう。

4-1. 問題の解説

AOJでは、各問題のページから、必要に応じて、ノート（解説）を開くことができます（解説がついている場合）。ノートは、問題文の前と後ろにその入り口が表示され、それぞれ問題文を読む前に役立つノート、問題文を読んだ後または問題を解いた後に役立つノートが開きます。

また、ノートは「言語解説」と「一般解説」のように分類され、言語解説では、選択したプログラミング言語の文法解説や模範解答を閲覧することができます。一般解説では、アルゴリズム、ヒント、関連リンクなど、プログラミング言語に依存しない解説や情報を閲覧することができます。

これらのノートの内容は、分類・対象を選択することで、必要なときに必要な内容を展開・表示することができます。例えば、以下の問題では、問題文の前にノートへの入り口が表示されています。



例えば、C++言語を選択して言語解説ボタンをクリックすると、以下のようにC++言語の前提知識に関する解説が展開します。

The screenshot shows a web browser window with the URL `https://onlinejudge.u-aizu.ac.jp/#/courses/lesson/2/ITP1/2/ITP1_2_A`. The page header includes the AIZU ONLINE JUDGE logo, navigation links for 'コース' (Courses), 'チャレンジ' (Challenge), and 'アリーナ' (Arena), and a user profile for 'Yutaka Watanobe'. The breadcrumb trail is 'Courses > Lesson - ITP1 > ITP1_2_A'. The problem details show 'Time Limit: 1 sec', 'Memory Limit: 131072 KB', and 'isSolved: [checked]'. A 'Note' section is visible with a 'C++' dropdown and a '言語解説' (Language Explanation) button.

条件分岐

関数のブロック内に書かれている文は基本的に上から順番に実行されますが、**構造文**と**条件式**を用いることで処理の内容を変えることができます。

処理の流れを制御する最も基本的な構造文は、条件によって異なる処理を実行することができる**条件分岐**です。条件分岐には **if 文**、**if-else 文**、**if-else if-else 文** などがあります。

「もしAなら1を実行する」はif文を用いて次のように記述します：

```
if ( A ){  
    1  
}
```

```
graph TD; Start(( )) --> Cond{条件式}; Cond -- true --> Proc[処理]; Cond -- false --> Join(( )); Proc --> Join; Join --> End(( ))
```

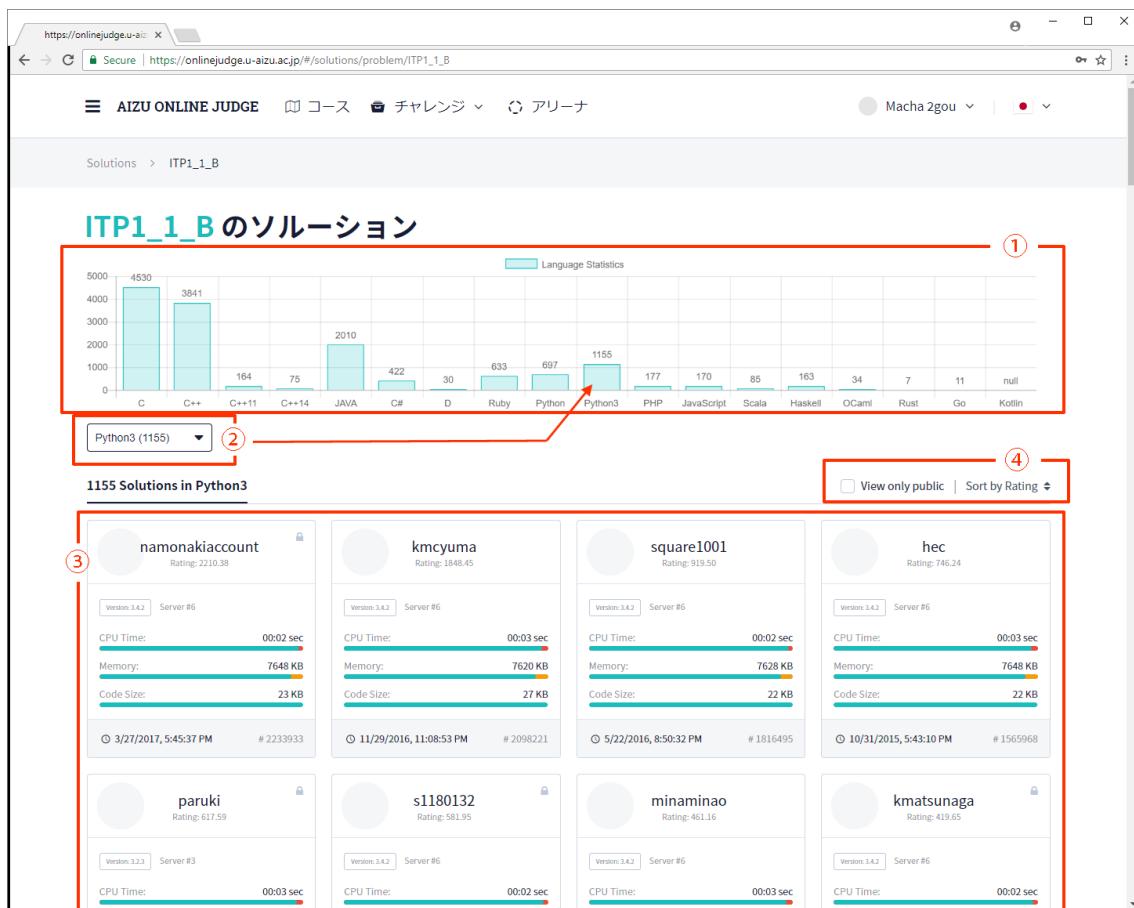
4-2. 公開されている解答ソースコード

AOJでは、ユーザ設定により、自分が提出したソースコードを公開することで、他のユーザの学習に貢献することができます。学習者の立場としては、公開されている他のユーザのコードを自由に閲覧することで、他のユーザから学ぶことができます。

ユーザのソースコードへは、主に以下のページからアクセスすることができます：

- 各問題の Solutions ページ
- 提出履歴のジャッジ ID リンク

問題文ページ下部フッタの Solutions リンクから、以下のような、各問題のソリューションページを開きます。



ページの上部 ① には、プログラミング言語ごとのソリューションの数がチャートで表され、チャートのバー、またはチャート左下のボックス ② からプログラミング言語を選択することで、特定言語のソリューション一覧を表示することができます。最初に開いたときは、ユーザ登録で設定されている、普段利用しているプログラミング言語のソリューションリストが表示されます。

表示エリア ③ には、ソリューションの一覧が表示され、右上のメニュー ④ から、表示順の基準を設定することができます。デフォルトでは、ユーザのレーティング順で表示されます。また、提出日時順、CPU・メモリ・コード長などの性能順でソートを行うこともできます。

各ソリューションはカードで表され、ソースコードの作者（ユーザ ID）に加えて、CPU 時間やメモリ使用量などの概要を確認することができます。

カードを選択すると、以下のように、ソースコードが表示されます。

The screenshot shows a web browser window displaying the AIZU ONLINE JUDGE website. The URL is <https://onlinejudge.u-aizu.ac.jp/#/solutions/problem/0582/review/1024526/dohatsu/C++>. The page shows a solution card for problem 0582, submitted by user DOHATSU. The card displays the source code in C++ and performance metrics for four submissions.

```

1 #include<iostream>
2 #include<algorithm>
3 using namespace std;
4 int t[3],x,y;
5 int a=0,b=0,c=0;
6 int main(){
7     while(1){
8         cin>>t[0]>>t[1]>>t[2];
9         sort(t,t+3);
10        x=t[0]*t[0]+t[1]*t[1];
11        y=t[2]*t[2];
12        if(t[0]+t[1]<t[2]){
13            break;
14        }else if(x==y){
15            a++;
16        }else if(x>y){
17            b++;
18        }else if(x<y){
19            c++;
20        }
21    }
22    cout<<a+b+c<<" <<a<<" <<b<<" <<<<<<endl;
23    return 0;
24 }
25

```

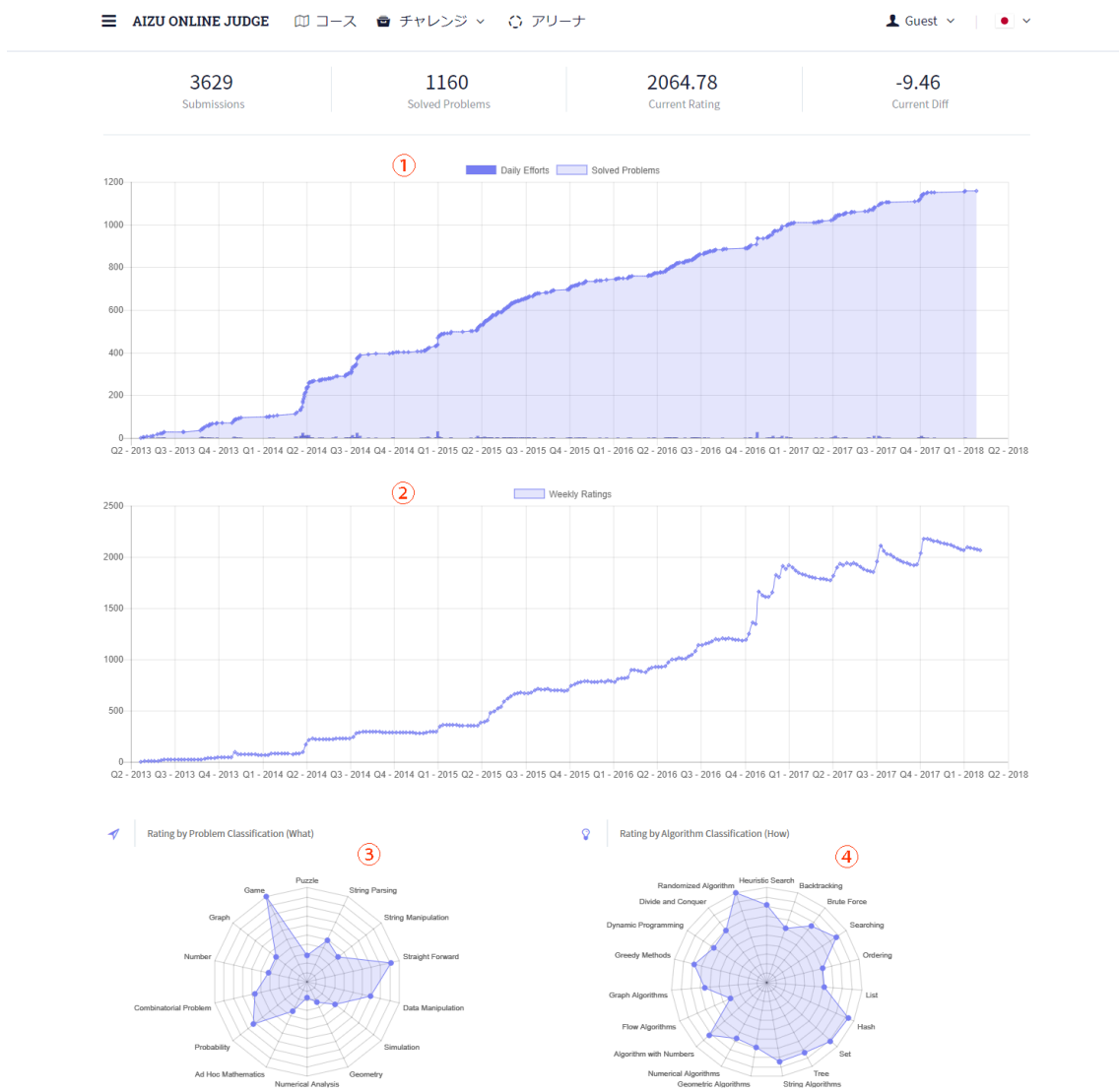
reviewed : 79

CPU Time	Memory
00:00 sec	1040 KB
00:00 sec	1040 KB
00:00 sec	1040 KB
00:00 sec	1184 KB

4-3. ユーザステータス

オンラインジャッジでは、自由なタイミングで好きな問題を解くことができますが、正解数などで競い合えば、時間制限のない長期的な競争ととらえることもできます。そこで、一般的なオンラインジャッジシステムでは、ユーザの活動や成績を数値化して公開することで、反復演習と習慣化を促しています。

AOJでは、コースとチャレンジそれぞれについて、全体あるいはトピック・セクションの達成度を確認することができます。



最初のチャート Solved Problems, Daily Effort ① は、日ごとの正解した問題数とその累計を表すチャートです。

次のチャート Weekly Rating ② は、一週間ごとのレーティングの推移を表します。ここでレーティングは、各問題に割り当てられたスコアの合計と考えてください。

最後のレーダーチャート Rating By Problem Classification ③、Rating By Algorithm Classification ④ はそれぞれ、問題の分類によるスキル、解法の分類によるスキルを可視化します。これらのレーダーチャートからは、おおよその得意・不得意分野を把握することができます。

4-4. ランキング

AOJ では、ランキング（メニュー→ランキング）にてユーザのランキングを公開しており、主に以下のカテゴリでランク付けを行っています。

- **Solved（問題数）**：オンラインジャッジシステムの伝統的なランキングです。コース問題とチャレンジ問題で正解した問題数でランク付けされます。
- **Rating（レーティング）**：レーティング = 正解した「チャレンジ問題」のスコアの合計でランク付けされます。
- **Leaderboards - Languages（プログラミング言語）**：プログラミング言語ごとに、「チャレンジ問題」と「コース問題」両方について、正解した問題数でランク付けされます。
- **Leaderboards - Contests（コンテストのソース）**：コンテストのソースごとに、「チャレンジ問題」について、正解した問題数でランク付けされます。

5. 関連ツール

現在、以下のサービスを提供しています。※開発中のため、インターフェースや機能が更新される可能性があります。

プログラミングアリーナ : <https://onlinejudge.u-aizu.ac.jp/beta/arena.html>

オンラインエディタ : <https://onlinejudge.u-aizu.ac.jp/services/ice>