競技プログラミングを始めよう

終に鮭

電子計算機研究会

December 2, 2020

自己紹介

- 終に鮭
- 情報系学科 1 回生
- 競プロ歴 8ヶ月 (最初のコンテストは 4/4 の ABC161)
  - AtCoder しかやってない
  - 現在のレートは 877
  - ごくまれに yukicoder の問題も解く
- プログラミング歴は 4〜5 年 (?)
  - 現在メインで使っている言語は Python, C++, JavaScript
  - 使ったことのある言語は C, Java, Visual Basic, PHP

今日の内容

1. 競技プログラミングとは？
   - 競プロのいいところ
   - コンテストの形式
   - オンラインジャッジ
2. AtCoder をやってみよう
   - 対応言語
   - 書き方と注意点
   - 提出の練習
   - 典型問題を解いてみよう
   - 解説
3. 本番のコンテストについて
   - 参加方法
   - おすすめの言語
   - 注意点
   - 役に立つツール

Section 1

競技プログラミングとは？

競技プログラミングとは

プログラミング早解きコンテストのこと !!

競プロのいいところ

- プログラムが書けるようになる
  - 化生の友人 (未経験) に AtCoder を教えたらプログラミング 1 が楽単になった
- アルゴリズム・データ構造を学べる
  - 次回があればこの部分を掘り下げて勉強会を開くかも
- 計算量を意識して書くようになる
  - 逆に、$1 \leq N \leq 10^6$ なら $O(N \log N)$ で 2 秒に間に合うな、とメタ読みできる
- 数学力がつく
- 就活に役立つことがある (らしい)
  - レートが実力の証明になる

コンテストの形式

参加者全員に同じ課題が与えられ、より早く要求を満たすプログラムを記述することを競う

- アルゴ：短時間、最適解が出せる
- マラソン：中長時間、最適解を目指す
- コードゴルフ：より短いコードを記述することを競う

オンラインジャッジ

- AtCoder(日本)：週 1,2 回、全問日本語に対応
- CodeForces(ロシア)：開催頻度がすごい
- TopCoder(アメリカ)：マラソンが充実してる
- Aizu Online Judge(日本)：JOI や ICPC の過去問が解ける

Section 2

AtCoder をやってみよう

対応言語

多すぎて TEX がクラッシュするので一部のみ抜粋 (独断で選びました)

主な対応言語（ABC183 時点）

- C / C++
- Java
- C#
- Rust
- Python / PyPy
- Cython
- JavaScript (Node.js)
- PHP
- Ruby
- Perl

延べ 57 言語が使用可能 [a]

[a] https://atcoder.jp/contests/abc182/rules

プログラムの書き方と注意点

- 「入力」は標準入力から受け取れる、「出力」は標準出力に出す
  - コマンドライン引数は使わない
  - 標準入出力を備えていない言語の場合はちょっと特殊（practice contest 参照）
  - 例えば JavaScript では `/dev/stdin` からファイル入力、`console.log` で出力
- (B 問題ぐらいから) 制約に注意する
  - 算術オーバーフロー
    - 黒魔法 `#define int long long`
  - 実行時間制限超過 (TLE : Time Limit Exceeded)
    - 1 秒間に処理できる量は 10^7 から 10^8 ステップ程度
  - スタックオーバーフロー
    - 再帰の制限は緩められる場合がある
  - `out_of_range`

提出の練習

- 常設中のコンテストから practice contest に飛ぶ
- 問題タブから「A - Welcome to AtCoder」を選んで解いてみよう
  - 自前のエディタで書くのがおすすめ
  - 事前調査で回答いただいた言語のサンプルをここに用意してあります

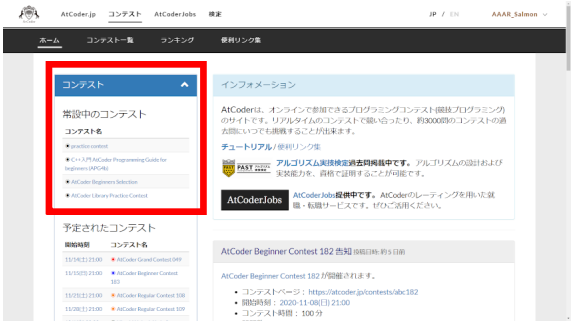

Figure: AtCoder のコンテストトップページ

提出の練習

- 必ずコードテストする
  - 手元の実行環境を使う
  - AtCoder のコードテスト機能を使う
- 問題ページ下部から提出

「いかにも競プロ」な問題を解いてみよう

- どんな問題？
  - 文字列操作系問題
  - 数学の問題
    - 小数の計算、幾何、方程式、数え上げ
  - データ構造を活かす問題
    - 配列、集合、スタック・キュー、Union-Find、全域木
  - 典型アルゴリズムを使う問題
    - ソート、探索、累積和、しゃくとり法、いもす法、動的計画法、貪欲法、最短経路問題、最大流問題

初心者におすすめの問題

- AtCoder Problems(有志サイト) にバーチャルコンテストを用意したのでやってみよう
- 制限時間は 1 時間
- 全問解ききることは想定していないので解けそうな問題から解いてください
- インターネットや本で調べても OK
- 全問解説すると時間がかかるので、全員解けた問題と聞いても誰もわからなさそうな問題は解説しません

解説

解説を始めます

配点基準

Table: 今回採用した配点の基準

配点	問題数	問題の要素
50	3	if 文のみで解ける 数学的考察が不要
100	4	if 文のみで解けるが、場合分けが難しい 文字列操作が必要 簡単な数学的要素が含まれる
200	7	配列を扱う ループまたは再帰関数が必要 単純な多重ループが必要 小数を扱う 数学的考察が含まれる

配点基準

Table: 今回採用した配点の基準

配点	問題数	問題の要素
300	3	ナイーブな実装が通用しない コーナーケースへの対応が必要
400	3	特殊なデータ構造を用いる 比較的複雑な典型アルゴリズムを用いる $\mathrm{mod}\ 10^9+7$ 複合的なスキルが必要

Section 3

本番のコンテストについて

参加方法

1. トップページの左側の「予定されたコンテスト」から選ぶ
   - ARC(橙)、AGC(赤) 系は難しいので ABC(青) 系のコンテストに参加しよう
   - レートが下がってもいいなら全部参加するのもあり
   - 現状は Nosub 撤退すれば Unrated になる
     - そのためにある程度解けてからまとめて提出する手法がある (tourist 出し)
2. 参加登録する
   - 企業コンの場合は個人情報を入力することも
3. 開催時間にもう一度開く

おすすめの言語

- C が書ける人：C++
  - 便利機能が多い
    - 文字列型、STL のコンテナクラスや関数、クラス、ラムダ式など
  - C 言語のコードはそのまま動く
  - 競プロでは一番人口が多い
  - 最近 ACL(AtCoder Library) が導入されて Union-Find などを実装しなくて良くなった
- ほとんど書けない人：Python
  - ライブラリが豊富で便利機能が多い
    - range 型、ジェネレータ、numpy モジュール
  - C と比べると遅いが、現状解けない問題は報告されていない
- 速度狂：Rust
  - C/C++より更に早い
  - 暗黙の型変換を一切しないのでめっちゃコンパイルエラーが出る
  - 全ての型の変数はデフォルトで immutable(書き換え不可)

注意点

- コンテスト開催中に Twitter などで余計なことは喋らないようにしよう
  - 全員に公開されている情報は OK
    - 「A 問題 AC した」は大丈夫、「A 問題 WA/TLE/RE した」はだめ
    - 「B 問題は余弦定理を使う」、「C 問題は bit 全探索で間に合う」などもだめ
    - ソースコードを上げたり生配信したりは当然だめ
  - 詳細はコンテストページ下部の「ルール」から確認できる
- 検索や自作ライブラリの使用は自由
  - ただしコンテスト中の情報共有は不可
  - コードスニペットやテンプレートを用意しておくとよい

役に立つツール

- AtCoder Problems
  - コンテスト終了後、問題の difficulty を推定して色で表示してくれる
    - API が提供されているので、AtCoder のページで色付けするスクリプトなども作られている
  - 過去問が参照しやすい
    - 解いたことのある問題が色付けされる
  - バーチャルコンテストが開催されている
- atcoder-cli
  - Node.js が必要
    - online-judge-tools が必要で、online-judge-tools の利用には Python3 が必要
  - ディレクトリの作成、実行テスト、提出がコマンドラインでできるツール
  - 問題を選択するとサンプルケースを拾ってきてコマンド一つでテストできる