競プロ作問を支える技術

@tsutaj

自己紹介

- 各種コンテストに tsutaj という名前で出ています
  - 最近は長期マラソン型コンテストによく出てます
- これまで、**競プロの作問準備**にしばしば関わってきました
  - 学生のとき：北海道大学の有志セット
  - 現在：ICPC OB/OG 会で提供する模擬国内予選・模擬地区予選
- 今回は作問作業で使われている技術の話をします

競技プログラミングの問題に必要なもの

- 問題文
- データセット（入力ファイル・想定解法の出力ファイル）
- ジャッジ用プログラム（問題によっては必要）
- 解説

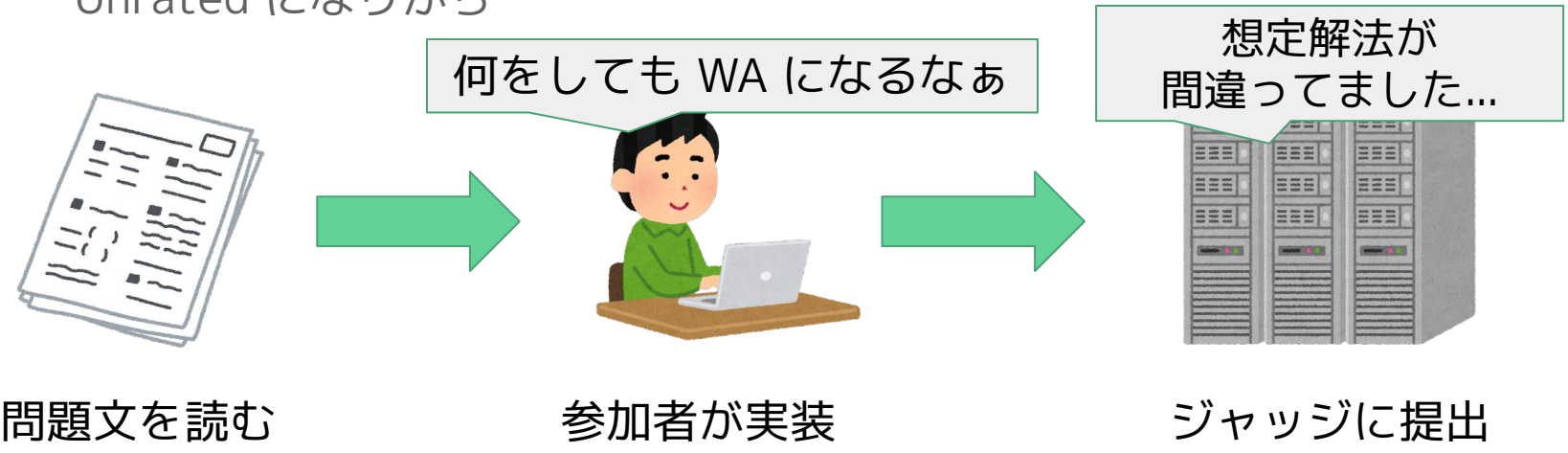
問題文を読む

参加者が実装

ジャッジに提出

よくあるミス 1

- 想定解法にミスがあり、正しい解法が通らない
  - コーナーケースを見落としていた、間違ったものを想定解としてセットしてしまったなど、原因は様々
  - 最もしてはいけない部類のミスで、レートが変動するコンテストの場合は Unrated になりがち



よくあるミス 2

- 問題文に書いてある制約と、データセットの制約が異なる
  - たとえば、想定よりも大きい入力が来ると、意図しない TLE や配列外参照のおそれがある

$N \leq 10^5$ だよ！

問題文を読む

`int A[100010]; …`
よし！提出したろ！

参加者が実装

$N \leq 2 \times 10^5$ のケースもありました…

ジャッジに提出

競技プログラミングの問題に必要なもの

- **齟齬がなく、内容が正確な**問題文
- **問題文の制約通りの**データセット
- **ジャッジ上で正確かつ高速に動く**ジャッジ用プログラム
- 解説

問題文を読む　参加者が実装　ジャッジに提出

競技プログラミングの問題に必要なもの

- **齟齬がなく、内容が正確な**問題文

コンテストを壊さないためには、
問題文・データセット・ジャッジプログラムに
正確性や再現性が求められる！

問題文を読む

参加者が実装

ジャッジに提出

主な作問支援ツール

- Rime
  - データセット作成支援
  - 想定解法・想定誤解法のチェック
    - ジャッジプログラムを含んだ問題のチェックもできる
  - ICPC OB/OG 会の有志が開発中
- statements-manager
  - 問題文作成支援
    - 問題制約を正確に書くために使用
  - tsutaj が開発中！

Rime

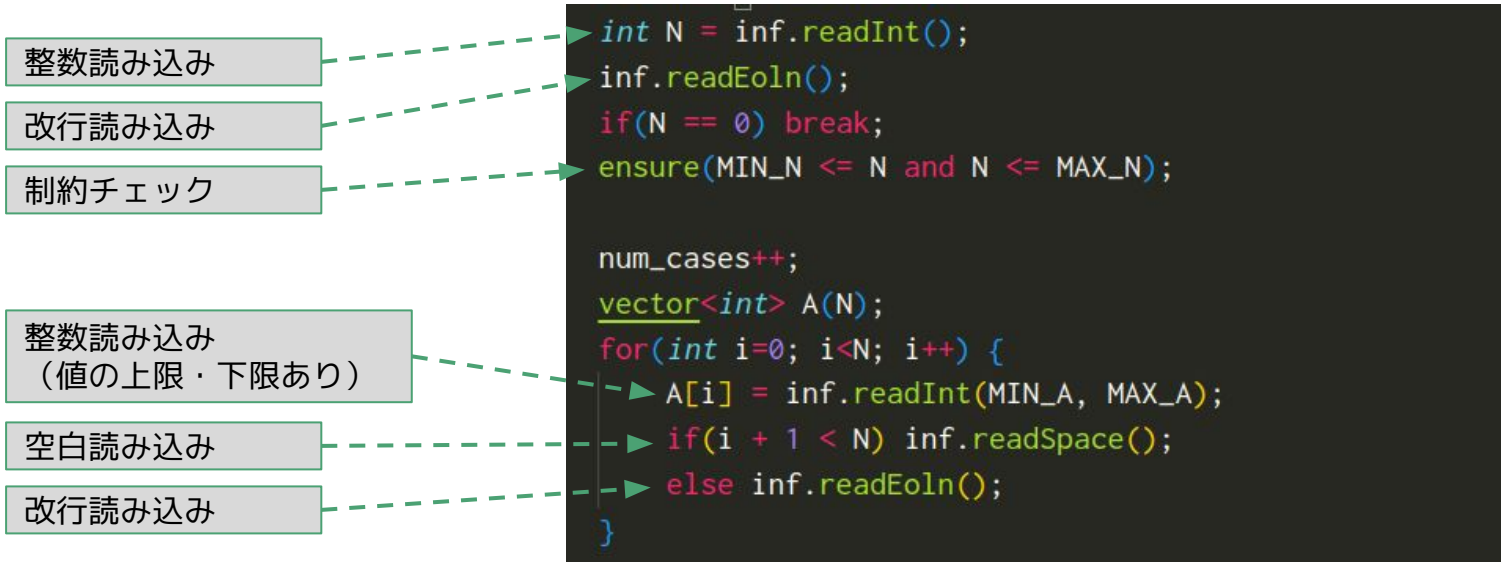
- できること
  - データセット**作成**を自動で回す
  - データセット**確認**を自動で回す
    - データセットは制約を満たしているか？
  - **解法プログラム**を自動で回す
    - 想定解が通って、想定誤解法が通らない状態か？
    - 解法は複数存在するのが望ましい

```
[ COMPILE  ] dev/tests: generator.cpp
[ COMPILE  ] dev/tests: gen_challenge.cpp
[ COMPILE  ] dev/tests: validator.cpp
[ GENERATE ] dev/tests: generator.cpp
[ GENERATE ] dev/tests: gen_challenge.cpp
[ VALIDATE ] dev/tests: OK
```

```
Test Summary:
dev ... 4 solutions, 41 tests
  tsutaj_AC1         OK  max 0.14s, acc 1.56s, score 100
  tsutaj_AC2         OK  max 0.24s, acc 2.07s, score 100
  tsutaj_subtask_1   OK  19_challenge_00.in: Time Limit Exceeded, score 25
  tsutaj_subtask_2   OK  20_case_01.in: Wrong Answer, score 50
```

testlib

- データセット確認には **testlib** というライブラリがよく使われる
  - 空白・改行が意図通り入っているか
  - データの値が制約通りか



testlib の注意点

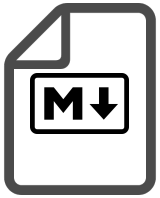
- ランダム入力を作る機能もありますが、オススメしません！
  - 線形合同法が使われているが、乱数の周期が短い
  - より周期の長いもの（最低限 XorShift）を使いましょう

- 余談：C++ の distribution も環境依存なので使うべきでない
  - 誰がプログラムを回しても同じデータが出ることが理想

statements-manager

- できること
  - 問題文に書かれる制約と、データセット作成に使われる制約を合わせられる
    - 問題文には $N \leq 10^5$ と書いてあるのに、データセットが $N \leq 2 \times 10^5$ を満たしているとヤバい
    - statements-manager で制約をまとめると、制約の情報を問題文でもデータセット作成でも使える
  - サンプルデータセットも同様に合わせられる

statements-manager



問題文
(Markdown)

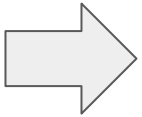
```
### 入力

```
$T$
$A_1$ $B_1$
$A_2$ $B_2$
...
$A_T$ $B_T$
```

- 入力は全て整数で与えられる
- ${@constraints.MIN_T} \leq T \leq {@constraints.MAX_T}$
- ${@constraints.MIN_AB} \leq A_i, B_i \leq {@constraints.MAX_AB}$
```

設定ファイル
(toml)

```toml
[constraints]
    MIN_AB = 1
    MAX_AB = 1_000_000_000
    MIN_T = 1
    MAX_T = 100_000
```



Problem A

A+B

2 つの整数 A, B が与えられます。$A + B$ を出力してください。

T 個のテストケースが与えられますので、T 行出力してください。

入力

T
$A_1\ B_1$
$A_2\ B_2$
...
$A_T\ B_T$

- 入力は全て整数で与えられる
- $1 \leq T \leq 100,000$
- $1 \leq A_i, B_i \leq 10^9$

制約が自動で置換される

出力

T 行出力してください。

i 行目には $A_i + B_i$ の計算結果を出力してください。

入力例

```
3
3 9
100 101
1 10
```

サンプルが自動で埋め込まれる

出力例

```
12
201
11
```

さいごに

- コンテストをミスなく準備するのはかなり大変
  - 今回のスコープ外ですが、問題文を齟齬なく書くのも大変
- 何らかの理由で Unrated になっても、次のコンテストを楽しみにして、また参加しましょう！
- 興味のあるかたは作問にも挑戦してみてください
  - yukicoder では有志コンテストがよく行われています