

競技プログラミング 基礎

16FMI えるざっぷ

Table

- ・ 競技プログラミングとは

- ・ 言語・ツール

基礎基礎知識

- ・ 計算量・アルゴリズム

- ・ ACM-ICPC コンテスト

- ・ 去年

コンテストについて

- ・ 学習方法

競技プログラミング とは

なぜやるのか

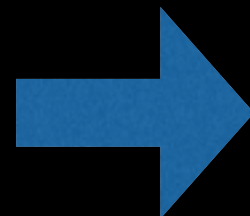
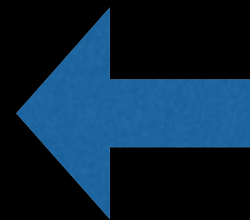
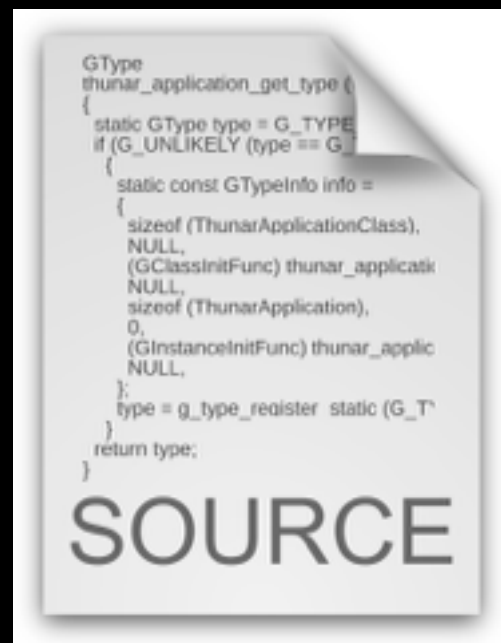
- ・ データ構造, アルゴリズムを学べる
- ・ 数学
- ・ 言語の(基礎部分の)勉強(x 綺麗にかけるようになる)
- ・ 競い合う楽しさ

なに

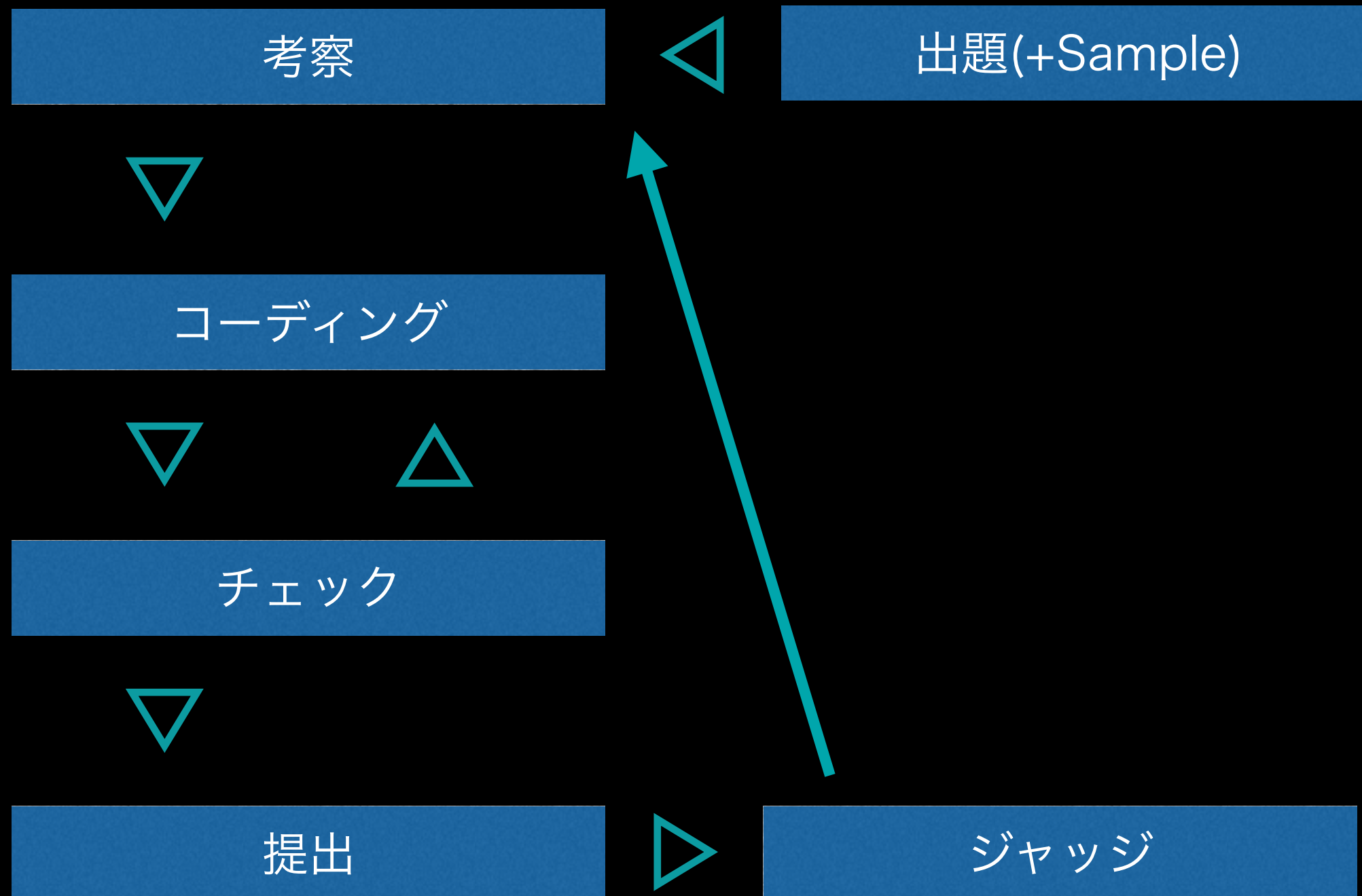
- ・ 「コーディングスピードと正確さ」を競う
- ・ 適切なアルゴリズムを選んで問題を解く

なに

- ・ 基本的な形式
- ・ 「A という input に対して処理した結果 B という output をするプログラムを作りなさい」



どうやって



・問題
・サンプル
(入力/正解出力)

考察



出題(+Sample)



コーディング



チェック

サンプル



ソース
コード



提出



ジャッジ

どうやって

- ・ 実演
- ・ <http://abc001.contest.atcoder.jp/>

言語・ツール

言語

- C++
- Java
- C, C#, Python3, Ruby

計算量

計算量

- Order: $O(\text{計算量})$
- 最悪計算量: 競プロではあらゆるテストケースに対応
- $O(n \log(n))$, $O(n!)$

計算量

少

- ・ $O(1)$ 定数時間
- ・ $O(\log(n))$ 対数時間
- ・ $O(n)$ 線形時間
- ・ $O(n^c)$ 多数項時間
- ・ $O(c^n)$ 指数時間

多

計算量

- ・ だいたい 1秒 $O(10^7)$
- ・ 例
 - ・ 入力値 a ($0 \leq a \leq 100000$)
 - ・ a 回ループの二重 for $\rightarrow O(a^2) \rightarrow O(10^{10})$
 - ・ 前処理してから $O(a)$ に落とす方法を考える

計算量

- ・ 例
- ・ 入力値 N, M ($0 \leq N \leq 100000, 0 \leq M \leq 10000$)
- ・ (NM) の計算はできない
- ・ $(N \log M)$ とかに出来ないか考える

計算量

- ・ 時間計算量: 処理回数 (←ここまでの話)
- ・ 空間計算量: メモリ

問題種類

- ・ アルゴリズム

種類

- ・ シミュレーション
- ・ 文章読解
- ・ 探索
- ・ 幾何
- ・ 再帰
- ・ 文字列操作
- ・ DP
- ・ グラフ

アルゴリズム

- ・ 総当り, 貪欲法, DP(動的計画法)
- ・ 二分探索, EulerTour
- ・ ダイクストラ, a^* , ワーシャルフロイド
- ・ UnionFind, LIT, いもす
- ・ ナップサック, メモ化
- ・ 逆元

学习方法

学習方法

- ・ Online コンテスト参加
- ・ Online Judge 問題, コンテスト過去問
- ・ 本

オンラインコンテスト

- ・ CodeForces(Div1, Div2)
 - ・ 競プロ界のステータス, 特殊
- ・ AtCoder
 - ・ 初心者向け
- ・ TopCoder

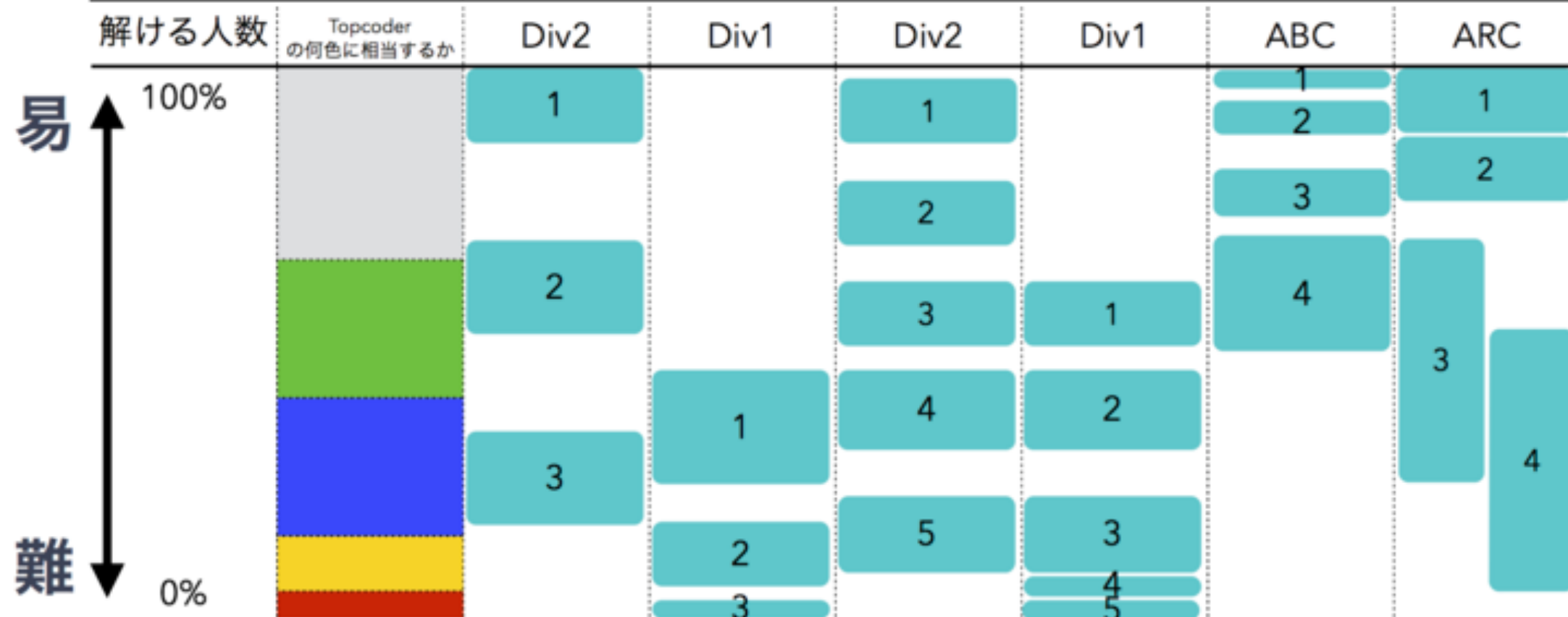
問題を解く

- ・ 過去問, コンテストはだいたい公式で公開している
- ・ 解説記事, 公式もある
- ・ 他の人の回答を見て学ぶ

主要なコンテストの紹介

初心者におすすめ

	TopCoder	Codeforces	AtCoder
開催頻度	月3回	月6回	毎週土曜日
問題文	英語	英語	日本語
問題数と時間	3問/75分	5問/120分	4問/90分
PythonのVersion	2	2, 3	2, 3



問題がある場所リスト

AOJ

judge.u-aizu.ac.jp

Introからある
アルゴリズム問題をやるべき

AtCoder

atcoder.jp

幅広い難易度
公式解説あり

Hackerrank

hackerrank.com

言語ごとの Intro からある

CodeForces

codeforces.com

難しめ, 英語

CodeIQ

codeiq.jp

アルゴリズム学べそうな
問題を選ばないと無駄

paiza

paiza.jp

アルゴリズム学べそうな
問題を選ばないと無駄

CodeFight

codefights.com

CodeGolf で
アルゴリズム学べたりする

本

・ 蟻本

ACM-ICPC

プログラミングコンテスト

ACM-ICPC

- ・ 国内予選（オンライン）：(日本語)
- ・ -> アジア予選 - 筑波：(英語)
- ・ -> 世界大会：(英語)

ACM-ICPC国内予選

- ・ 簡単なルール
- ・ 3時間, 2016/6/24(Fri)
- ・ 言語: C++, Java

ACM-ICPC国内予選

- ・ チームプレイ
- ・ 3人で1チーム
- ・ PC: チームで1台

ACM-ICPC国内予選

- ・ 環境
 - ・ インターネット不可
 - ・ 問題, 提出ページのみ可
 - ・ 紙類, 資料持ち込み可

ACM-ICPC国内予選

- ・ ジャッジ方法
 - ・ 実行時間の制限なし
- ・ 解答(Output) ファイルとソースを同時に提出
- ・ -> 計算量の数十倍になる程度なら無駄な処理も無視できる
- ・ -> x 一つ指数が上がるようなのは無理

- ・ 問題
- ・ サンプル
(入力/正解出力)
- ・ 問題入力

考察

出題(+Sample)

コーディング

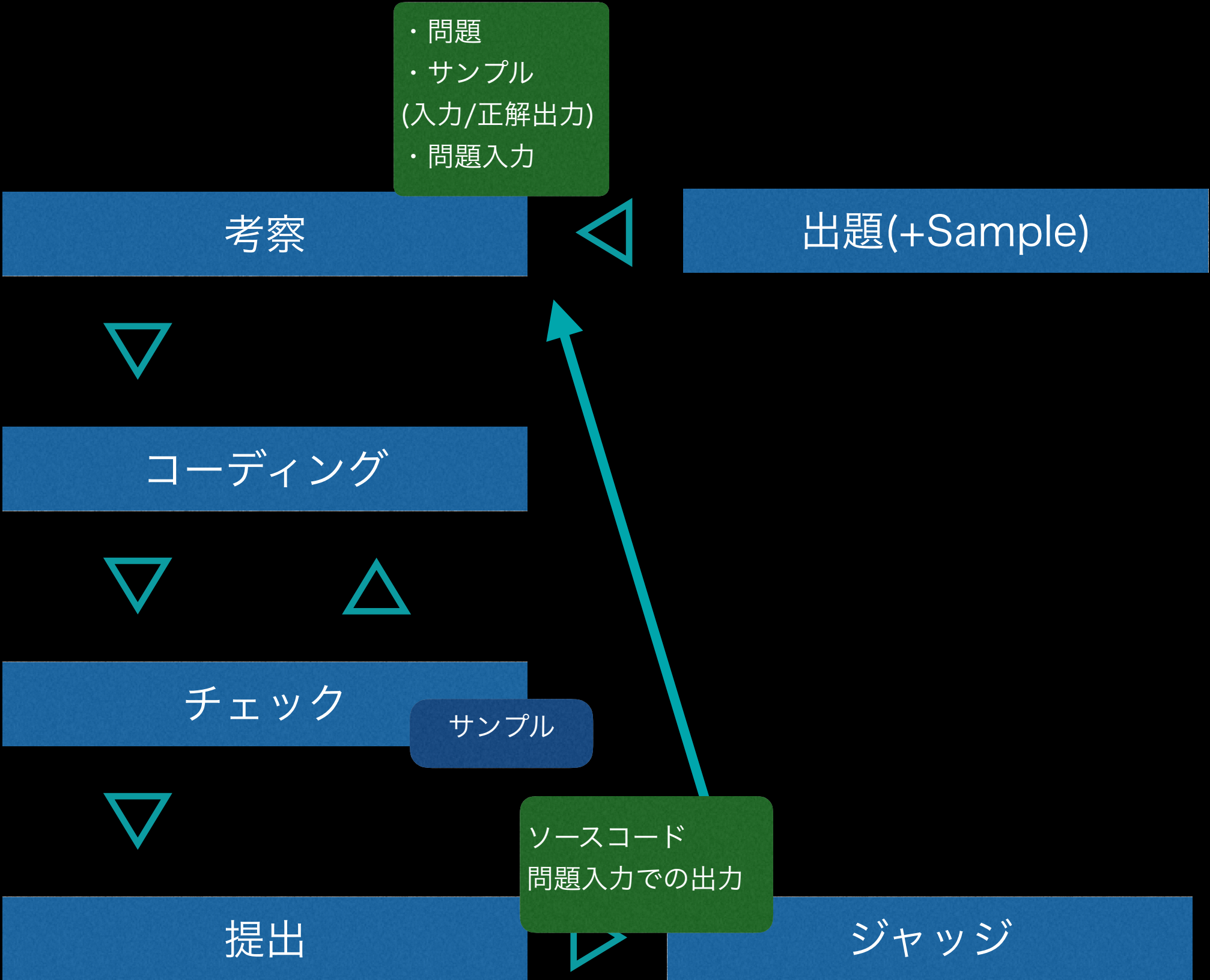
チェック

サンプル

提出

ソースコード
問題入力での出力

ジャッジ



去年体験記



千葉君, えるざっふ, KP

去年体験記

- ・ コンテストに向けた準備
- ・ 勉強会なし…
- ・ 俺がやってみたくて誘った

去年体験記

- ・ 本戦前
 - ・ 大量に大きなメモ用紙持ち込み
 - ・ コード, アルゴリズムのテンプレートを印刷持ち込み
 - ・ アルゴリズム本持ち込み

去年体験記

- ・ 本戦開始
- ・ 問題一気に印刷
- ・ 問題を読みながら誰がどれをやるのかを決める
- ・ 一人はコーディングを始める

去年体験記

- ・ A問題: 総当り, 解けた
- ・ B問題: シミュレーション, 解けた
- ・ C問題: ノード, ギリギリ解けた
- ・ 3完答- 終了

去年体験記

- ・ 通過チーム
 - ・ 7問正解 1チーム
 - ・ 6問正解 1チーム
 - ・ 5問正解 10チーム (2 同学校制限)
 - ・ 4問正解 14チーム (11 同学校制限)
 - ・ 3問正解 11チーム (以下90敗退)
- ・ 東京大学, 会津大学, 京都大学, 大阪大学, 筑波大学, 東京工業大学…

去年体験記

- ・ 反省点
 - ・ エディタ3人分用意するべき
 - ・ 担当する問題の種類をしっかりと選ぶ

参考文献

- ・ オーダー: <http://www.slideshare.net/catupper/ss-26238956>
- ・ プロコン: <http://www.slideshare.net/iwiwi/wakate-web-14323842>