競技における諸注意

2012/03/19

一般的な注意

- 競技時間は長いです
  - 落ち着いて取り組みましょう
    - 早く解くコンテストではありません
- 競技時間は長いです
  - 水分補給・トイレ休憩は適切に
    - 気分転換としての利用

問題を読む

- 問題は全部読みましょう
  - 難易度順に並んでいるとは限らない
  - 「簡単な問題を読んでいなかった」は痛い
- Overview Sheet もちゃんと見ましょう
  - 時間制限やメモリ制限から問題の雰囲気を察することができるかもしれない？

問題を読む

- 読み落とし・読み間違いは防ぐ
  - 読み間違いは非常にもったいないです
  - 慣れている人でもよくやります
  - 読んでいるうちに重要な条件を忘れがち
    - 重要そうなことはどんどんメモしましょう
  - サンプルを手で解いて確認
- 解釈に不安があれば質問しましょう
  - 損にはなりません

アルゴリズム

- 解法が思いついたら，実装する前に
  - 実装できそうか考える
  - 実装にかかる時間を見積もる
  - よりよい，楽な方法がないか少し考える
  - サンプルをそのアルゴリズムに従って解いて確認

アルゴリズム

- 明記された部分点に頼る
  - 得点になる
  - 部分点のアルゴリズムがヒントになる
  - 入力のどのパラメータが重要か
- 明記された部分点に頼りすぎない
  - 実装時間との兼ね合い
  - 小さいサイズだと可能な操作が多い
    - 自然な部分点解法がわからないまま満点解法がわかってしまう，なんてことも

プログラム

- 必ず丁寧にテストをしましょう
  - ケアレスミスによるバグはつきもの
  - 入出力例やフィードバックは必ず確認
  - 自分でテストケースを作る
    - コーナーケース
      - 小さいケースをたくさん作り，愚直な解法と比較
    - 大きいケース
      - どういうときに時間やメモリがかかるかを考える
  - 複数ケースに対して実行できるようにプログラムを書く方が便利？ (個人差あり)

プログラム

- コンパイラを頼りましょう (個人差あり)
  - g++ -Wall -O2 hoge.cpp -o hoge
    - g++ -Wall -Wextra -O2 hoge.cpp -o hoge も
- デバッガを使える人は頼りましょう
- いわゆる printf デバッグも有効
  - fprintf(stderr, "%d¥n", __LINE__);
    fflush(stderr);
  - cerr << __LINE__ << endl;

プログラム

- 満点解法の正当性や実装に自信がないときは

```
if (N <= 3000) {
    /* O(N^2) で解く */
} else {
    /* O(N log N) で解く */
}
```

– もちろん境界の値はテストして決める

提出

- 提出先は正しいか確認
  - 特に終了直前
- 提出するコードが最新のものか確認
- ときどきメモつきでバックアップをとるとよいかも

暇になったと感じたら

- 改めて問題文を丁寧に読む
  - 誤植を探す勢いで
- ひたすらテスト
- プログラムを全部目で読み，意図通りに書けているか確認
- 定数倍高速化 (時間超過しうる場合)
  - ただしバグが入らないように

おしまい

- IOI 目指してがんばってください