

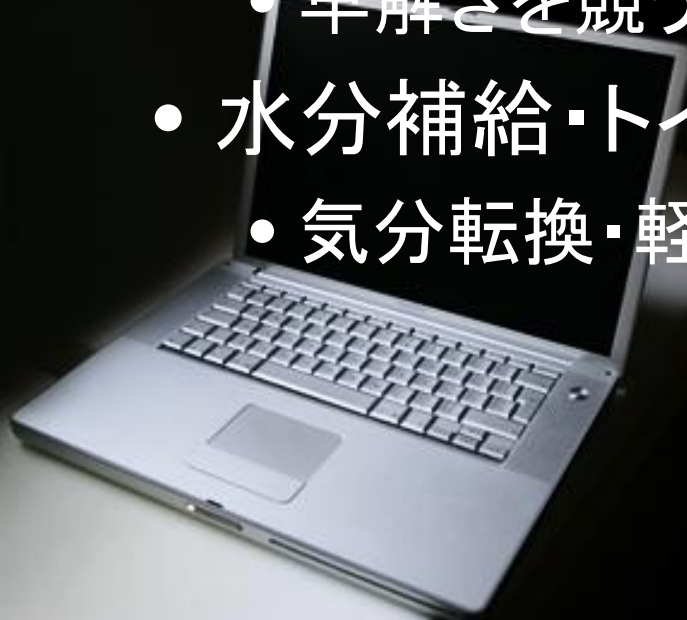
# 競技における諸注意

2014/03/19



# 一般的な注意

- 競技時間は**長い**です
  - 競技前の**体調管理**を怠らないように
  - 落ち着いて取り組みましょう
    - 早解きを競うコンテストではありません
  - 水分補給・トイレ休憩は適切に
    - 気分転換・軽い散歩としての利用



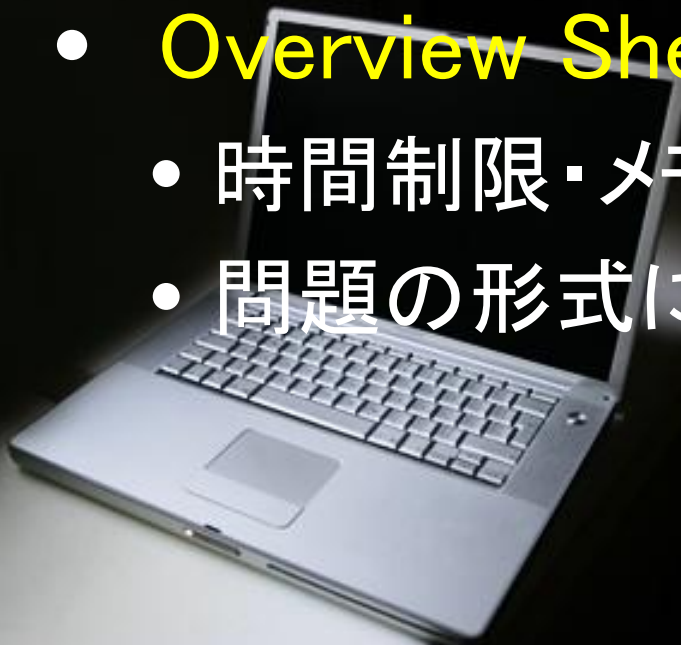
# 一般的な注意

- 合宿期間も長いです
  - 合宿中の体調管理を怠らないように
  - 1日くらい大きな失敗をしても挽回は可能



# 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りません
  - 「簡単な問題を読んでなかった」は痛い
- **Overview Sheet** もちゃんと見ましょう
  - 時間制限・メモリ制限を勘違いしない
  - 問題の形式にも注意



問題を読む

- 問題は**全部**  
読みましょう



# 問題を読む

- 読み落とし・読み間違いは防ぐ
  - 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
    - 重要そうなことにはメモ・印をつける
- 入出力例を手で解いて確認



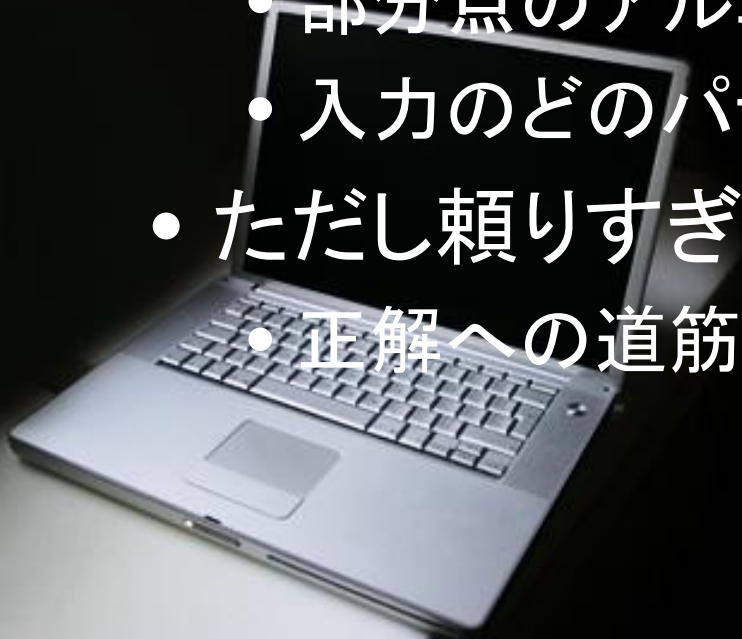
# 問題を読む

- 解釈に不安があれば早めに質問しましょう
  - 損にはなりません



# アルゴリズム

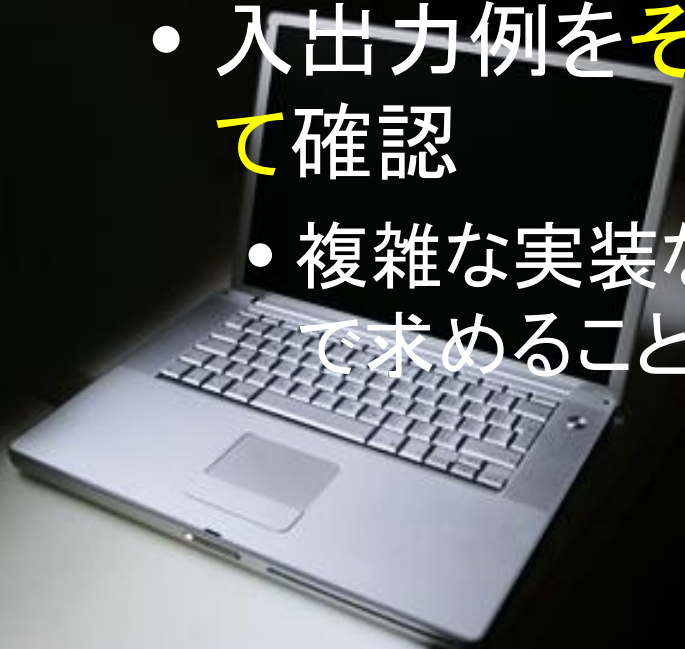
- 解法が思いつかないとき
  - 明記された**部分点・小課題**に頼る
    - 得点になる
    - 部分点のアルゴリズムがヒントになることも
    - 入力のどのパラメーターが重要かわかることも
  - ただし頼りすぎないこと
    - 正解への道筋が複数あるかも





# アルゴリズム

- 解法が思いついたとき
  - 実装にかかる時間を見積もる
  - より楽な・確実な・簡潔な方法はないか
  - 入出力例をそのアルゴリズムに従って解いて確認
    - 複雑な実装なら入出力例に対してプログラムで求めることになる主要な値を手計算



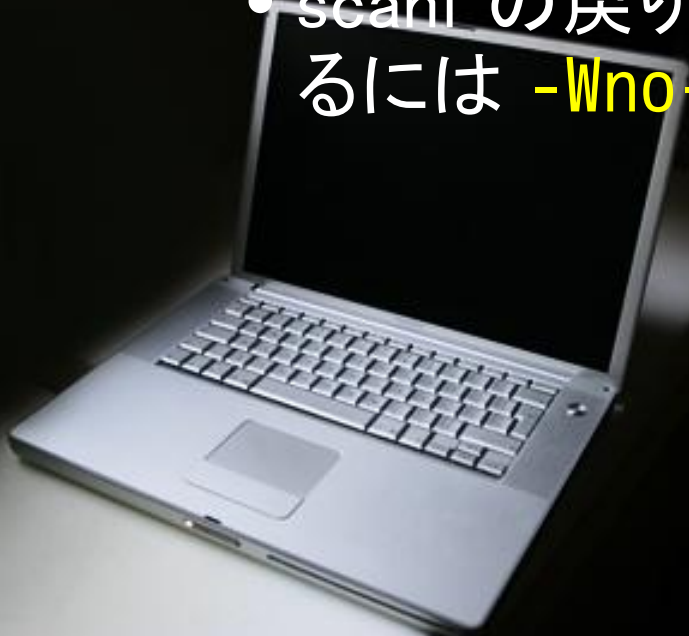
# 実装・デバッグ

- なるべく**慣れている**コードを書きましょう
  - 不慣れな処理を書く場合はその部分がデバッグしやすいように
- 複雑になりそうならまずは高速化せず**ナイーブ**に実装してみるのもあり



# 実装・デバッグ

- 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`
  - `scanf` の戻り値を使っていないことを無視させるには `-Wno-unused-result`



# 実装・デバッグ

- gdb などの**デバッガ**が使える人は頼りましょう
  - Segmentation Fault する場所がわかったり



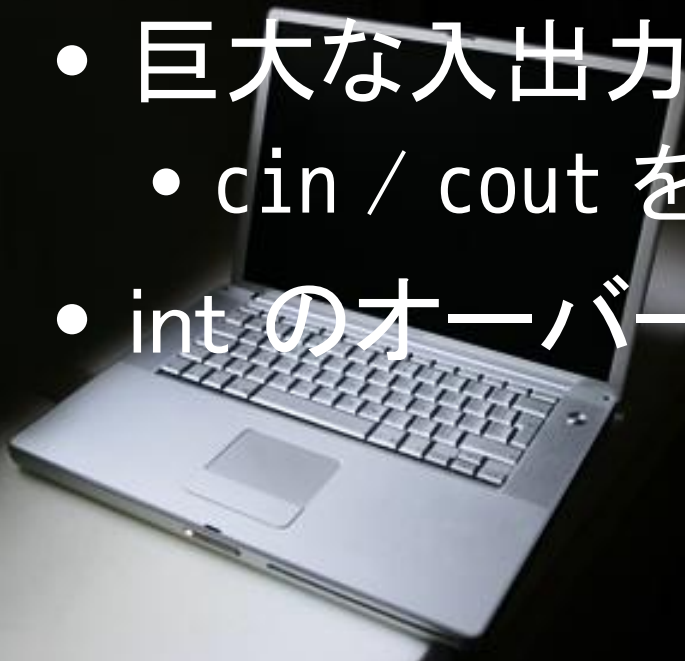
# 実装・デバッグ

- いわゆる printf デバッグも有効
  - `fprintf(stderr, "%d¥n", __LINE__);`  
`fflush(stderr);`
  - `cerr << __LINE__ << endl;`
  - 複数ファイルを実装するときはファイル名  
`__FILE__` も便利かも



# 実装・デバッグ

- **Technical Info Sheet** にも注目
- 手元実行での**スタックオーバーフロー**
  - `ulimit -s unlimited`
- 巨大な入出力
  - `cin / cout` を `printf / scanf` に
- `int` の**オーバーフロー** ( $2.1 \times 10^9$ )



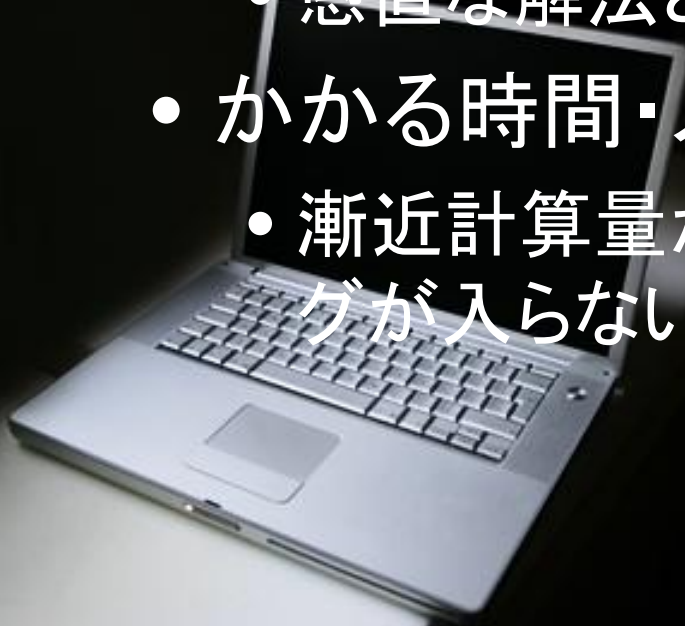
# 提出

- 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**をとるとよいかもしれません



# テスト

- フィードバックが通らないとき (or フィードバックが完全でないとき)
  - **小さいケース**をたくさん作る
    - 愚直な解法と比較
  - かかる時間・メモリが**大きいケース**を考える
    - 漸近計算量が大丈夫なのに通らないなら (バグが入らない程度に) 定数倍高速化





# テスト

- 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と便利？（個人差あり）
  - `while (~scanf("%d", &N)) { ... }`
  - 変数の初期化を忘れないように



# 暇になったと感じたら

- もう解けない！もうデバッグできない！
  - とれる**部分点**は本当にすべてとりましたか？



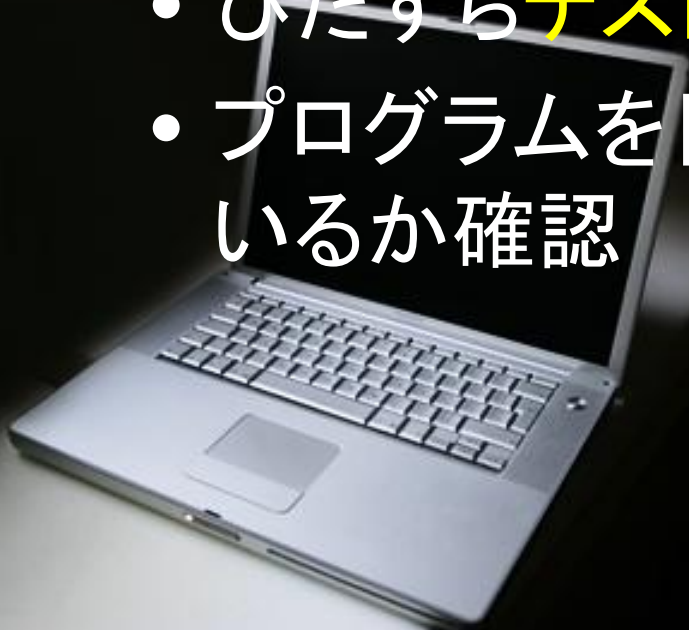
暇になったと感じたら

- とれる**部分点**は  
本当にすべてと  
りましたか？



# 暇になったと感じたら

- もう解けない！もうデバッグできない！
  - 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**してみる
  - プログラムを目で読んで意図通りに書けているか確認



# 暇になったと感じたら

- もう解けない！（もう満点！的な意味で）
  - 規則に違反しないように・周囲の迷惑にならないようにリラックスしててください
    - コードを綺麗にしてみるもよし
    - より速い解法を考えてみるもよし
    - 競技環境に慣れてみるのもよし



# おしまい

- IOI 目指してがんばってください

