CS3233
Competitive Programming

Dr. Steven Halim

Week 01 – Introduction

Outline

- Course Administration
  - Break 1, Clicker Distribution (~ 6.30-6.45pm)
- Competitive Programming Book, Chapter 1
  - Competitive Programming: **Live Demo**
  - Tips to be Competitive: **Hands on ☺, join me**
  - Break 2 (~ 7.50-8.00pm)
- Mooshak: First **Mock Contest** & Discussion
  - 45 minutes contest: 3 “easy” problems

CS3233 Lecturer History

- Initiated by **Prof Andrew Lim** (now CUHK): 1999-2001
  - Vacuum in AY 2002/03… ☹
- Between 2004-2006, CS3233 was taught by **A/P Leong Hon Wai** and **A/P Ooi Wei Tsang**
  - Another vacuum in AY 2007/08… ☹
- Revived again* on semester 2, 2008/09 ☺
- Note: Each lecturer has different style…
  - Mine is geared towards **ICPC preparation**

SoC Teams Performance History (1)

- ACM ICPC World Finals
  - 1999: Joint-18
  - 2000: Joint-22
  - 2001: Joint-29
  - 2003: Joint-13
  - 2005: Melvin, Junbin, Yunsong: Hon. Mention
  - 2009: Duc, Tien*, Phong: Hon. Mention
  - 2010: Duc, Tien*, Phong: Hon. Mention
  - 2011: Miss out by 2 ranks ☹
  - 2012: Zi Chun*, Harta*, Phuong* ☺
  - 2013: You?

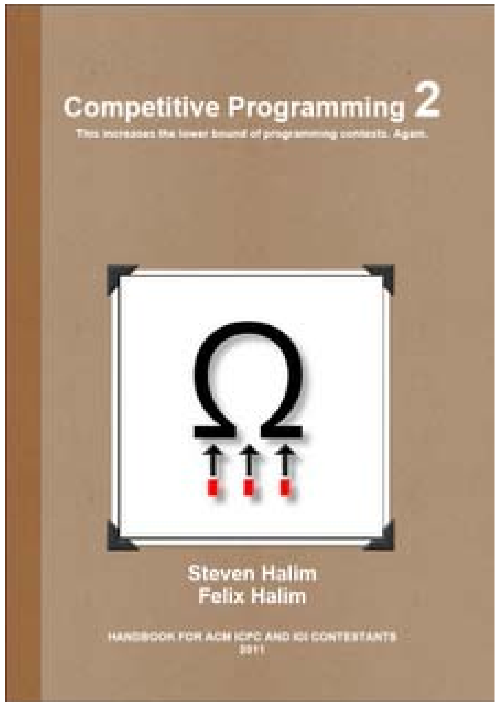
SoC Teams Performance History (2)

- Recent ACM ICPC Regional Contests
  - 2008: 6th in Amritapuri; 3rd in Kanpur; Joint-15th & Joint-16th in Kuala Lumpur
  - 2009: 7th & 10th in Jakarta; 3rd in Manila; 2nd and 10th in Phuket
  - 2010: 10th in Daejeon; 6th++ in Kuala Lumpur; 10th in Tokyo
  - 2011: 7th++ in Phuket; 5th++ in Kuala Lumpur
  - 2012: YOUR TURN for World Finals 2013!
- More history in:
  - http://algorithmics.comp.nus.edu.sg/wiki/

SoC Current Strengths

- Teaching Staffs and Seniors:
  - A/P Tan, Dr Steven, Duc, Tien*, Phong, Felix*, Su Zhan*, Suhendry, Victor, Zi Chun*, Harta*, Phuong*, etc
    - * ex/current-World Finalists currently in SoC
  - Singapore IOI Teams 2010-2011
    - 2 Golds, 2 Silvers, and 4 Bronzes by this team over the past 2 years
- Current Students:
  - Many potential students (YOU ALL)…

Textbook


Competitive Programming 2
This increases the lower bound of programming contests. Again.
Ω
Steven Halim
Felix Halim
HANDBOOK FOR ACM ICPC AND IOI CONTESTANTS
2011

- Competitive Programming 2
- COMPULSORY!!
- 25 SGD/copy
- Buy tonight (15 copies are available)
- (Private) lecture material is not uploaded…
  - Reason: To keep NUS advantage over other universities ☺
- Public version, without the starred slides, is in: http://sites.google.com/site/stevenhalim/home/material

Clicker Distribution (15 Mins Break)

- During the break:
  - Distribute clickers
  - Discuss administrative issues with me (i.e. should I take this module or not?, etc)
    - In NUS, drop with ‘W’ grade is after Week 02!
  - Buy “Competitive Programming 2” textbook

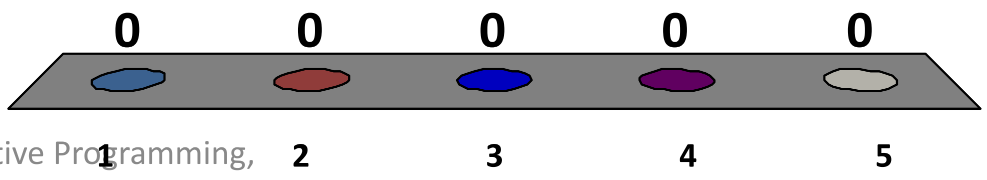
Time Check:
6.30pm

This increases the lower bound of programming contests. Again.

COMPETITIVE PROGRAMMING 2

How many of you have read CP2?

1. I have just bought it…
   So 0 page so far…
2. Only a few pages so far
3. Most of chapter 4 and a bit of chapter 3 due to CS2020/CS2010
4. I have read most of it, but I know you have other tricks not written in CP2 yet ☺
5. I have mastered CP2 **and beyond** ☺



0 of 120

Competitive Programming

- Given <u>well-known</u> Computer Science problems, solve them <u>as fast as possible</u>!
  - Not about “software engineering”
- Well-known = not research problems!
  - Problems in our target contest: ACM ICPC & IOI have this characteristic!

Demo (UVa 11849 – CD)

- This exaggerated demo illustrate contestant’s type:
    A. The blurry one
    B. Give up
    C. Slow
    D. Competitive programmer
    E. Very competitive programmer

TIPS TO BE COMPETITIVE

Tip 1: Type Fast & Correct

- No kidding, this can be important!
- Let’s try
  - http://www.typingtest.com
    - ZEBRA – Africa’s Striped Horse
  - Mine: ~85-90 wpm
  - Felix’s: ~55-65 wpm
- Familiarize yourself with the positions of the following keyboard keys:
  - (,),{,},[,],<,>,’,”,&,|,!,etc

Tip 2: Quickly Identify Problem Types

- Ad Hoc
- Complete Search
- Divide and Conquer
- Greedy
- Dynamic Programming
- Graph
- Mathematics
- String Processing
- Comp. Geometry
- Some Harder Ones

Tip 3: Do Algorithm Analysis

- This is taught in more details in CS3230!
- In this module, we will just learn the basics required for dealing with ICPC/IOI problems
  - See the constraints in the problem statement
  - Conjure the simplest algorithm that works!
  - Do some basic analysis to convince that it will work *before* we start coding…

Tip 4: Master Programming Language<u>s</u>

- You should master at least one (**preferably more**) programming language**s**
  - Reduce the amount of time looking at references
  - Use shortcuts, macros, avoid comments
  - Use libraries whenever possible
- Idea: Once you figure out a solution for a problem, you are able to translate it into a <u>bug-free</u> code, and do it <u>fast</u>!

Tip 5: Master the Art of Testing Code

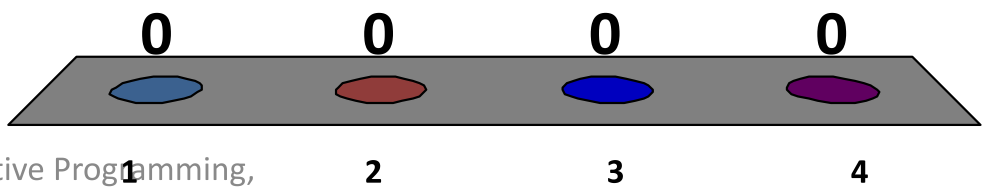
- Ultimately, we want “Accepted (AC)” verdict ☺
  - i.e. Our code passes the judge’s secret test data
- However, we may instead be given: ☹
  - Presentation Error (PE)
  - Wrong Answer (WA)
  - Time Limit Exceeded (TLE)
  - Memory Limit Exceeded (MLE)
  - Runtime Error (RTE)

Tip 6: Practice and More Practice

- Online Judges for CS3233
  - MAIN: University of Valladolid (UVa) Online Judge
    - http://uva.onlinejudge.org (Open with Firefox!)
  - MISC: ACM ICPC Live Archive
    - http://livearchive.onlinejudge.org/
  - MISC: TopCoder
    - http://www.topcoder.com
  - MISC: USACO
    - http://train.usaco.org

We will mainly use UVa online judge this semester. I have…

1. Registered a free account but have not solve anything
2. Have solved ≥ 10 UVa problems
3. Have solved ≥ 40 UVa problems
4. Have solved ≥ 100 UVa problems



0 of 120

Tip 7: Team Work (ICPC Only)

- Practice coding on a blank paper
- Submit and print strategy
- Prepare test data challenges
- The X-factor

Let’s start

THE AD HOC PROBLEMS

Ad Hoc Problems (1)

- Definition from USACO + modifications
  - 'Ad Hoc' problems are those whose algorithms do not fall into standard categories with **well-studied solutions**
  - Each Ad Hoc problem is **different**...
    - No specific or general techniques exist to solve them
  - This makes the problems the 'fun' ones (and sometimes frustrating), since each one presents a new challenge
  - The solutions might require a novel data structure or an unusual set of loops or conditionals

Ad Hoc Problems (2)

- Definition from USACO + modifications (Cont)
  - Sometimes they require special combinations that are rare or at least rarely encountered
  - It usually require careful problem description reading and usually yield to an attack that revolves around carefully sequencing the instructions given in the problem
  - Ad Hoc problems can still require reasonable optimizations and at least a degree of analysis that enables one to avoid loops nested five deep, for example

Ad Hoc Problems (3)

- Ad Hoc problems usually appear in ACM ICPC problem set (1 or 2 per set)
  - Can be of “general type” (listed in Chapter 1), using simple Data Structures (Chapter 2), Mathematical (Chapter 5), String-related (Chapter 6), or Basic Geometry (Chapter 7)
- Sadly, solving only Ad Hoc problems will not give your team a good result in ICPC…
  - We will learn more problem types in subsequent classes

Ad Hoc Problems (4)

- We further sub-divide Ad Hoc problems into the following sub-categories:
  - Super Easy; Easy; Medium
  - Game (Card); Game (Chess); Game (Others), Easier; Game (Others), Harder (more tedious)
  - Josephus; Palindrome; Anagram
  - Interesting Real Life Problems, Easier; Interesting Real Life Problems, Harder;
  - Time; ‘Time Waster’
  - Just Ad Hoc
  - Ad Hoc problems in other Chapters (2, 5, 6, 7)

Next Week

- **CH2: Data Structures and Libraries**
  – Focus on bit manipulation and Binary Indexed (Fenwick) Tree

10 Minutes Break

- In the last part of our first introductory class, you will familiarize yourself with **Linux controlled environment** in this PL6 and the **Mooshak system**, the internal online judge used for CS3233 this semester
  - Mock contest containing 3 "simple problems"
  - Not graded yet ☺, enjoy it for fun
    (and to help you decide if you should take CS3233)

Time Check:
7.50pm

Mooshak

- Let’s try this system
  - Open with **Firefox**
    (does not work well with most other browsers ☹):
    - http://algorithmics.comp.nus.edu.sg
  - Click “Online Judge (Login)” at the top left corner