Course Introduction

Dr. Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science

Welcome to CS 491 CAP!

Your Objectives:

- Describe the goals and prerequisites of this course.
- Describe the grading scheme.
- Be able to practice effectively.

Why take this course?

- Primary course goal: make you good at competitive programming!
- Why should you want to do that?
  - It's fun!
  - Opportunity to learn:
    - useful data structures, algorithms, and mathematical insights;
    - practical applications of data structures and algorithms;
    - how to code and debug effectively; and
    - how to work well on a team.
  - You'll do really well on job interviews!

Am I ready?

Course Prerequisites

- CS 225, CS 173, CS 125.
- We won't enforce this, but you'd better be ready to learn!

Skills Needed

-
- Proficiencey in programming C, C++, or Java (CS 125)
- Familiarity with basic data structures (CS 225).
- Comfortable with recursion and algorithmic explanations (CS 173).
- Most important: eagerness to learn and practice!!

Textbook *Competitive Programming 3* by Steven and Felix[**Halim2013a**]

SIG ICPC Team

- Preparing for 2019 Mid-Central ICPC Regionals
  - Will discuss and collaboratively solve problems from this seminar's problem sets
- Mailing list:
  - Join us!
  - `https://www-s.acm.illinois.edu/cgi-bin/mailman/listinfo/icpc-l`

Programming Contests

- UIUC ICPC tryouts and practice
  - One Local
  - One online
- ACM ICPC
  - Mid-central Regionals in Chicago (November 9 most likely)
  - World Finals
- Online contests
  - TopCoder SRMs, CodeForces
  - Facebook Hacker Cup
  - Google Code Jam
  - TopCoder Open
  - ... and many others ...

Online Judges

- Real contest problems
- Immediate Feedback
- Can emulate contest environment
- List of online judges:
  - UVa Online Judge `https://uva.onlinejudge.org/`
  - Peking Online Judge `http://poj.org`
  - ACM ICPC Live Archive `https://icpcarchive.ecs.baylor.edu/`
  - Sphere Online Judge (SPOJ): `http://www.spoj.com/`
  - Open Kattis `https://open.kattis.com/`
  - Saratov State Online Judge: `http://acm.sgu.ru/`
- **Get an account on each of these!**
- But... we will primarily use UVa this semester. We will send you a link to collect your online judge IDs later.

Online Contests

- Occur 3–4 times per month.
- Top Coder Single Round Matches (SRMs).
  `https://www.topcoder.com/`
- Code Forces
  `http://codeforces.com/`

UIUC ICPC Team Meetings

- SIG ICPC Website: `http://icpc.cs.illinois.edu/ipl.html`
  - Contains announcements, practice summaries, and practice resources.
- Meeting Calendar: `http://icpc.cs.illinois.edu/calendar.html`
- **Tryouts**
  - Two of them!
  - Dates to be announced....
- Practice contests on subsequent Saturdays.
- Details on http://icpc.cs.illinois.edu/calendar.html

Class Organization and Assignments

- Each period will have the following workflow:

  Lecture Video A short lecture video will introduce the topic.

  Sample Problem(s)
  - These will be posted to the web page.
  - The problem should be solved before class.
    - Put your solution into your git repository.
    - Be ready to discuss your solution. The instructor will anonymously post code for the class to view.
  - In Class problem — if there is time, we will solve a problem in class.

  Problem Set You will also get a "weekly" problem set.
  - Problems will be rated by difficulty: Easy, Medium, Hard
  - Problems should be submitted on corresponding online judge.

NB: Please do not copy-paste code from other sources. You are only hurting yourself if you do!

Grading

- Course is Pass/Fail: Passing is 70%.
- Attendance is worth 10%.
- Participation is worth 10%.
  - Measured by submission of practice problems for discussion.
  - You get four "excused absences" for both attendance and participation.
- Completion of problem sets is worth 80%.
  - Difficulty levels:
    - Easy problems: 1 point — straightforward application of algorithm
    - Medium problems: 3 points — nontrivial modification of algorithm needed to solve
    - Hard problems: 5 points — insight beyond the use of the algorithm may be needed
  - Completion of a problem set involves solving 6 points worth of problems.
  - If you took CS 491 CAP before, then *you may not use "easy" problems towards your completion!*
  - Due within two weeks of assignment. **No Extensions**
  - We will drop two problem sets. But really, you should do them all.

Extra Credit

There are opportunities for extra credit here too!

- Attending a tryout counts as one problem set.
- You can get points by contributing new problems to our problem sets.

Approach to Solving ICPC Problems

1. **Read the problem statement carefully!**
   - Pay attention to the input/output format specification.
2. Abstract the problem.
3. Design an algorithm.
4. Implement and debug.
5. Submit.
6. AC!
   - (else `GO TO` 4... or maybe even 3)

Example Problem

- POJ 1000: A + B Problem
  - Input: two space separated integers, a and b.
  - Constraints: $0 \leq a, b \leq 10$.
  - Output: $a + b$

C / C++ Code for POJ 1000

```c
#include <stdio.h>

int main() {
  int a, b;

  scanf("%d %d", &a, &b);
  printf("%d\n", a + b);
  return 0;
}
```

Java Code for POJ 1000

```java
import java.io.*;
import java.util.*;

public class Main {
  public static void main(String args[])
  throws Exception{
    Scanner cin=new Scanner(System.in);
    int a=cin.nextInt(), b=cin.nextInt();
    System.out.println(a+b);
  }
}
```

Example Problem

- POJ 1004 — Financial Management
  - Input: 12 floating-point numbers, each on a separate line
  - Output: Average of the numbers, rounded to two decimal places
  - Note that the answer must be preceeded by a dollar sign ($)!

C/C++ Code for POJ 1004

```c
#include<stdio.h>

int main() {
  double sum = 0, buf;
  for(int i = 0; i < 12; i++) {
    scanf("%f", &buf);
    sum += buf;
  }
  printf("$%.2f\n", sum / 12.0);
  return 0;
}
```

Java Code for POJ 1004

```java
import java.util.*;

class Main {
  public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    double d = 0;
    for (int i = 0; i < 12; ++i) {
      d += in.nextDouble();
    }
    System.out.printf("$%.2f\n", d/12.0);
  }
}
```

Questions?

Course Resources

- Course Website: `https://pages.github-dev.cs.illinois.edu/cs491cap/web-fa19`
- Mailing list: `https://www-s.acm.illinois.edu/cgi-bin/mailman/listinfo/icpc-l`
- Piazza page: (NO solution posts!) `https://piazza.com/class/jzio8t35i4y5u4`
- UIUC ICPC team website: `http://icpc.cs.illinois.edu/`
- Announcements will be sent to the ICPC mailing list and put on Piazza
- Course materials will be available on the website
- UVa Online Judge: `https://onlinejudge.org`
- uHunt (UVa Problem Hunting Tool): `https://uhunt.onlinejudge.org/`

Bibliography