Programming Contest problems

Dr. Jeyakesavan Veerasamy
CS faculty, The University of Texas at Dallas
Email: jeyv@utdallas.edu
Website: www.utdallas.edu/~jeyv
Welcome to CodeWarmers!

Jey Veerasamy & John Cole
CS faculty, UT Dallas
Pre-requisites

• Participants should be comfortable with the basics of programming in Java/C++. Those basic concepts are decisions, loops, functions/methods, and arrays.

• If you are new to programming, you should complete CS1336 Programming Fundamentals course in UT Dallas, or CSK12OutreachUTD workshops OR300 Java/OR301 C++ first.
Logistics

• You should bring a laptop with WiFi access to the session.

• Each session will have ~1 hour of lecture followed by ~2 hours of hands-on problem-solving.

• You should have a working compiler & IDE environment for C++ / Java. If you have one already, you can skip the next 2 slides.
Java compiler & IDE

www.oracle.com/technetwork/java/javase/downloads

- **Compiler**: Java Platform (JDK) 7u11
- **Simple IDE**: jGRASP [www.jgrasp.org](http://www.jgrasp.org)
  (or)
- **Complex IDE & bundle**: JDK 7 + NetBeans
  (or)


Eclipse IDE for Java Developers
C++ compiler & IDE

• MS Visual Studio / Visual C++ Express
  (or)
• codeblocks-12.11mingw-setup.exe from www.codeblocks.org/downloads/26
  (or)
• NetBeans C/C++ bundle from netbeans.org/downloads/

Google for “Configuring the NetBeans for C++” and follow steps to install a C++ compiler.

(or)
www.eclipse.org/downloads/
Eclipse IDE for C/C++ Developers
IOI/USACO

• International Olympiad in Informatics
• USA Computing Olympiad (USACO) feeds to it
• Information & competitions: www.usaco.org
• USACO training site: ace.delos.com/usacogate
  http://cerberus.delos.com:790/usacogate
• We need to create accounts in both sites.
IOI/USACO – 2012-13 Schedule

• Online programming competitions 6 times a year – open to all

Nov 16-19: November Contest
Dec 14-17: December Contest
Jan 11-14: January Contest
Feb 8-11: February Contest
Mar 8-11: March Contest
April 5-8: US Open
Late May/Early June: Training Camp
July 6-13: IOI'13 in Brisbane, Australia
ACM ICPC

- ACM International Programming Contest Information: icpc.baylor.edu
- ACM programming competitions training site: uva.onlinejudge.org
ACM ICPC Competitions

- College level
- Region level
- International Competition: ACM-ICPC World Finals, June 30 – July 4, 2013, St. Petersburg, Russia

- UTD has a group called Codeburners – Dr. Ivor Page trains the team – Programming competition scholarships are available.
More online competition sites

- www.codechef.com
- www.topcoder.com
- www.spoj.com
- projecteuler.net

Lot more at
en.wikipedia.org/wiki/Category:Programming_contests
USACO competitions

• Individual competitor
• Uses input files & output files (NO standard input/output)
• Upload and submit the source file for assessment
• Typically 10 testcases
• For each case, result can be YES, NO, or Time-over
• Need to get OK for all testcases
• Helps you by pointing out invalid output & showing the correct output – for practice only!
ICPC competitions

- team based (each team has 3 college students)
- Uses input files & output files (NO standard input/output)
- Varying # of testcases
- Upload and submit the source file for judging
- 3 indications: Success, Wrong answer, or Time Exceeded
- No information on test input/output provided for wrong answers – even during practice!
Programming Competitions
vs.
Commercial Application Development

• Algorithms
• Classes and other OOP concepts
• Coding style & comments
include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main() {
    ofstream fout ("beads.out");
    ifstream fin ("beads.in");

    fin >> x;

    ... fout << x;
    ... fout.close();
}
Start-up code for Java

/*
ID: jeyak71
Needed for USACO training site
PROG: beads
LANG: JAVA
*/
import java.util.*;
import java.io.*;

class beads {

    public static void main(String[] args) throws Exception {
        Scanner input = new Scanner(new File("beads.in"));
        PrintWriter output = new PrintWriter("beads.out");

        int x = input.nextInt();

        output.println(x);
        output.close();
    }
}
Problem set for Session #1

- USACO training site: Your Ride is Here
- USACO site Nov 2012 contest: Find the Cow!
- USACO site Dec 2012 contest: Meet and Greet
- USACO training site: Friday the Thirteenth
- USACO training site: Broken Necklace
PROBLEM NAME: cowfind
INPUT FORMAT:
Line 1: A string of parentheses of length N (1 <= N <= 50,000).

SAMPLE INPUT (file cowfind.in):
) ( ( ( ) ( ) ) ( )

OUTPUT FORMAT: * Line 1: The number of distinct pairs of indices x < y at which there is the pattern (( at index x and the pattern )) at index y.
SAMPLE OUTPUT (file cowfind.out):
4
Why 4?

) (( ( ( ) ( ) ) ( ) )

) (( ( ( ) ( ) ) ( ) )

) (( ( ( ) ( ) ) ( ) )

) (( ( ( ) ( ) ) ( ) )

) (( ( ( ) ( ) ) ( ) )

) (( ( ( ) ( ) ) ( ) )
Sample C++ Code #1

```cpp
for(int i=0 ; i<s.length()-1 ; i++)
    if ((s[i] == '(') && (s[i+1] == '('))
        for(int j=0 ; j<s.length()-1 ; j++)
            if ((s[j] == ')') && (s[j+1] == ')'))
                count++;
```
Results for Submission #1

What is the problem?
for(int i=0 ; i<s.length()-1 ; i++)
    if ((s[i] == '(') && (s[i+1] == '('))
        for(int j=i+2 ; j<s.length()-1 ; j++)
            if ((s[j] == ')') && (s[j+1] == ')'))
                count++;
Results for Submission #2

Almost there! Need to be more efficient.
More efficient?

- Input string length can go up to 50000
- Nested loop takes long time!
- Brute force algorithm $\rightarrow$ more intelligent algorithm
More efficient?

If we look at the sample input,

) ( ( ( ) ( ) ) ( ) )

we can keep track # of open (( combinations, when )) is seen, we can add that many (( options to count!
```cpp
int count = 0;
int openCount = 0;

for(int i=0 ; i<s.length()-1 ; i++) {
    if ((s[i] == '(') && (s[i+1] == '('))
        openCount++;
    else if ((s[i] == ')') && (s[i+1] == '))
        count += openCount;
}
```
Results for Submission #3

1. * 11.3mb < 1ms
2. * 11.3mb < 1ms
3. * 11.3mb 2ms
4. * 11.3mb < 1ms
5. * 11.3mb 1ms
6. * 11.3mb 1ms
7. * 11.3mb 1ms
8. * 11.3mb 1ms
9. * 11.3mb 1ms
10. * 11.3mb < 1ms
Thank you!

• My website: www.utdallas.edu/~jeyv
• USACO: www.usaco.org
• USACO training: ace.delos.com/usacogate

• Few solutions: www.utdallas.edu/~jeyv/compete

Questions & Answers?