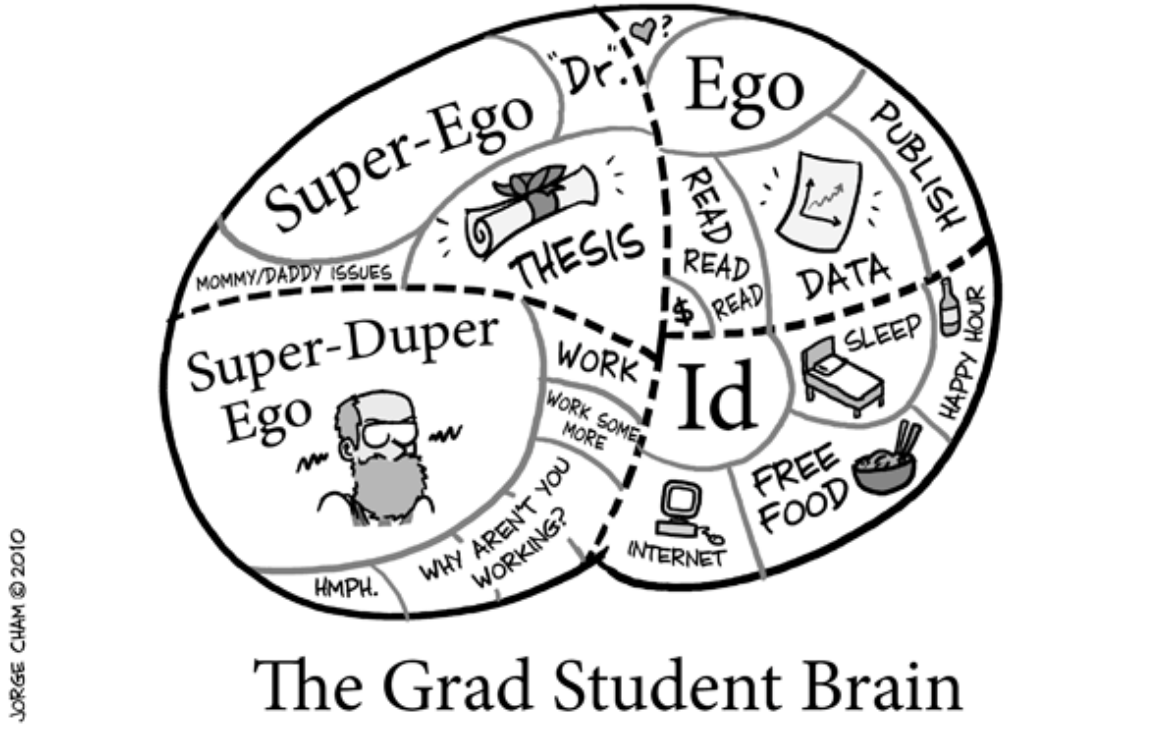
Competitive Programming

An Introduction

Paul Grigoras
paul.grigoras09@imperial.ac.uk

About Me


Super-Ego
"Dr."
Ego
PUBLISH
THESIS
READ
READ
READ
$
DATA
MOMMY/DADDY ISSUES
Super-Duper Ego
WORK
Id
SLEEP
HAPPY HOUR
WORK SOME MORE
FREE FOOD
INTERNET
WHY AREN'T YOU WORKING?
HMPH.
JORGE CHAM © 2010
The Grad Student Brain
WWW.PHDCOMICS.COM

First Attempt…

Learning Outcome

- Solve interesting problems
- See
  - new tools
  - new languages
  - cool tricks

`$whatis` Competitive Programming

- solve & implement 3 - 12 algorithmic problems
- 1 - 5 hours
- individual or teams of 2 - 3
- limited run-time ~1s
- limited memory ~large enough
- 100% correct (no edge case bugs)

`$whereis` Competitive Programming

- UKIEPC http://www.cs.nott.ac.uk/~mlw/ukiepc/2013/
- ICPC Regionals - Northwestern Europe
- ICPC World Finals
- Online:
  - codeforces
  - topcoder

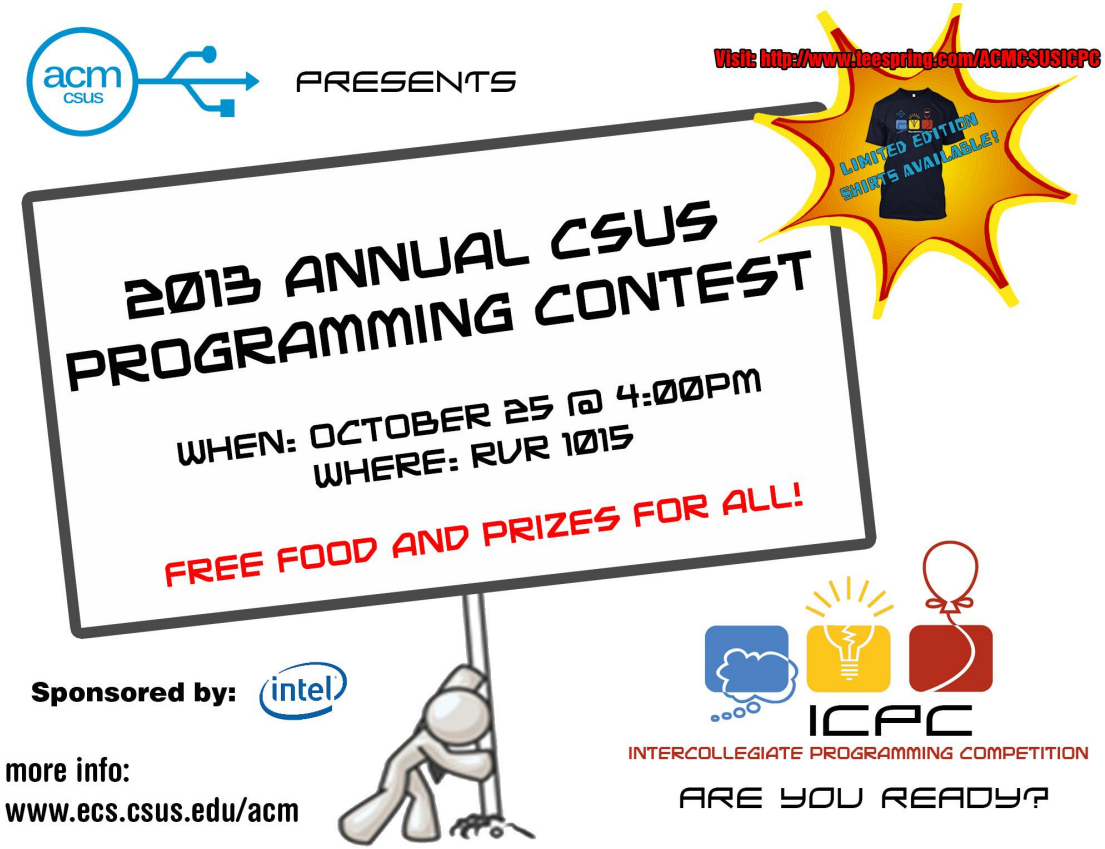
Competitive Programming@Imperial

- Winter Contest @ Imperial
  - 1st February - Sign Up NOW!

- Workshops with past ICPC contestants
  - starting 29th of January
  - Christian Ledig christian.ledig@imperial.ac.uk
  - practice: www.doc.ic.ac.uk/icpc

Why - The Dreaded Coding Interview

Why - Fun


acm
csus
PRESENTS
Visit: http://www.teespring.com/ACMCSUSICPC
LIMITED EDITION
SHIRTS AVAILABLE!
2013 ANNUAL CSUS
PROGRAMMING CONTEST
WHEN: OCTOBER 25 @ 4:00PM
WHERE: RVR 1015
FREE FOOD AND PRIZES FOR ALL!
Sponsored by:
intel
more info:
www.ecs.csus.edu/acm
ICPC
INTERCOLLEGIATE PROGRAMMING COMPETITION
ARE YOU READY?

Why - Really Useful

- Great Practice
  - Problem Solving
  - Complexity Analysis
  - Coding
  - Use APIs
- For 2nd, 3rd, 4th year courses

How - Online Judges

- amazing problem archive - UVA
- neatly structured - uhunt
- instant feedback
- great for practice

How - Online Contests

- real-time contests (up to 1-2k contestants):
  - Codeforces (almost weekly)
  - TopCoder (Prestigious - TopCoder Open)
- usually two divisions (pros and amateurs:))
- some are really prestigious:
  - Facebook Hacker Cup
  - Google Code Jam

How - Books

- *Competitive Programming*, Steven Halim
- *Introduction to Algorithms*, CLRS
- *The Algorithm Design Manual*, Skiena

Who

- Mainly Beginners
  - no competitive programming experience

- But also
  - people that want coding interview-like practice
  - people that need a refresher before going back to competitive programming

Overview

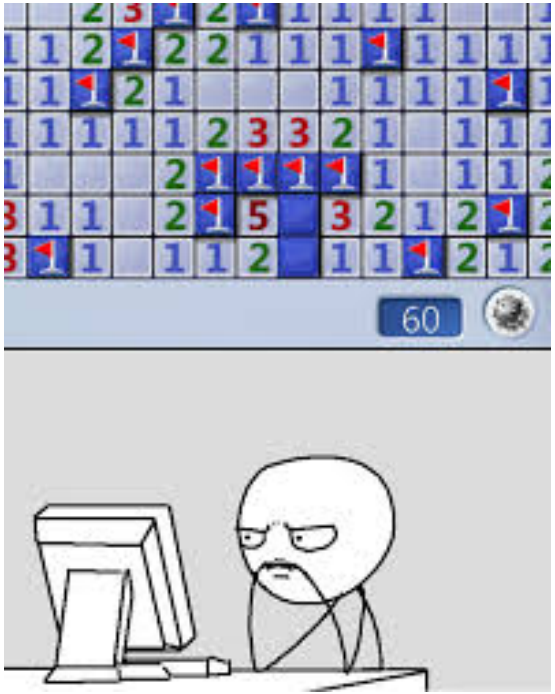
Implementation	This Week...
Complete Search	Week 2
Divide and Conquer	Week 3
Greedy, Dynamic Programming	Week 4
Graphs	Weeks 5 & 6
Maths	Week 7
Geometry	Week 8

Languages

- **Haskell** - cool, but rarely available on OJs
- **Java** - not so cool, but available (*we'll use it*)
- **C++** - even better (but you don't know it.. yet)

1 Implementation

Minesweeper



Minesweeper [UVA 10189](#)

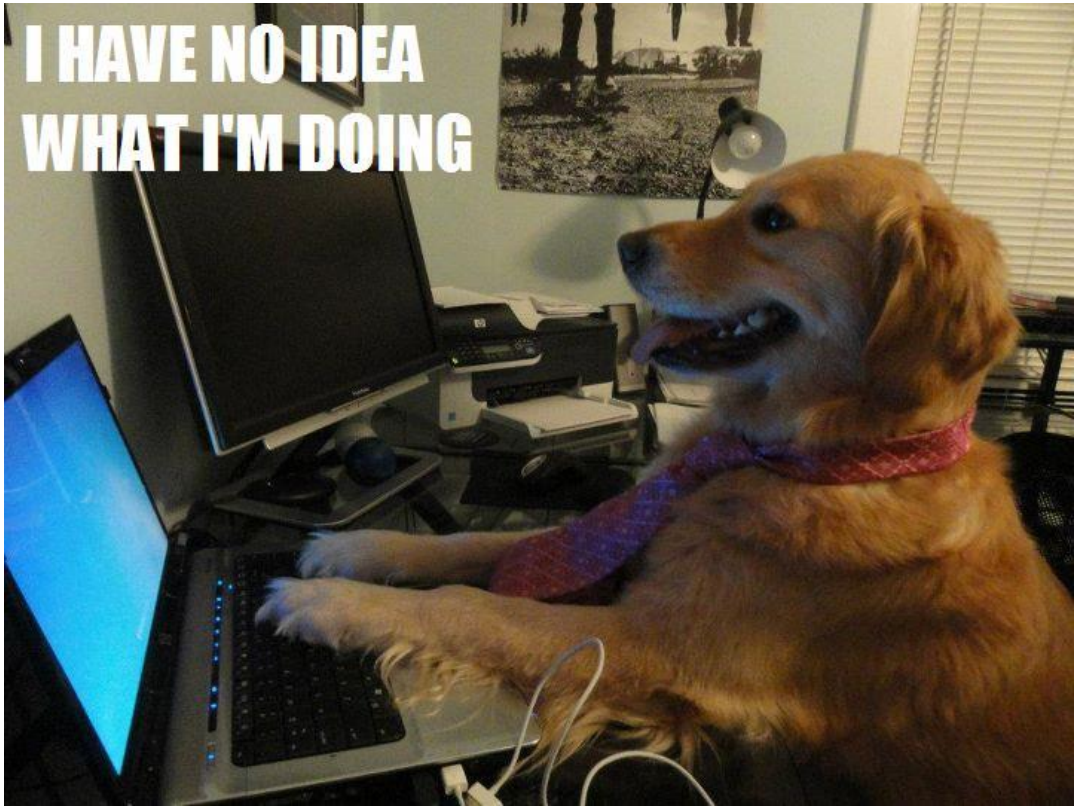
Problem statement on UVA

Count how many mines are adjacent to each tile of an N X M minesweeper map.

```
4 4
*...
....
.*..
....
```

```
*100
2210
1*10
1110
```

Minesweeper - Solution



Analyze the Problem

Problem statement:

1. What is it asking
    a. try examples on paper
2. What is the I/O format
    a. this often tricks people and is really frustrating
3. What are the time/memory constraints
4. What are the input constraints

Minesweeper - Solution

Pretty easy:

1. Keep a count matrix
2. Loop through the board
   - if we find a mine (*) at i, j
     i. increment count for all non-mine adjacent fields
3. Print the count matrix

Minesweeper - Solution

Think before you write any code!

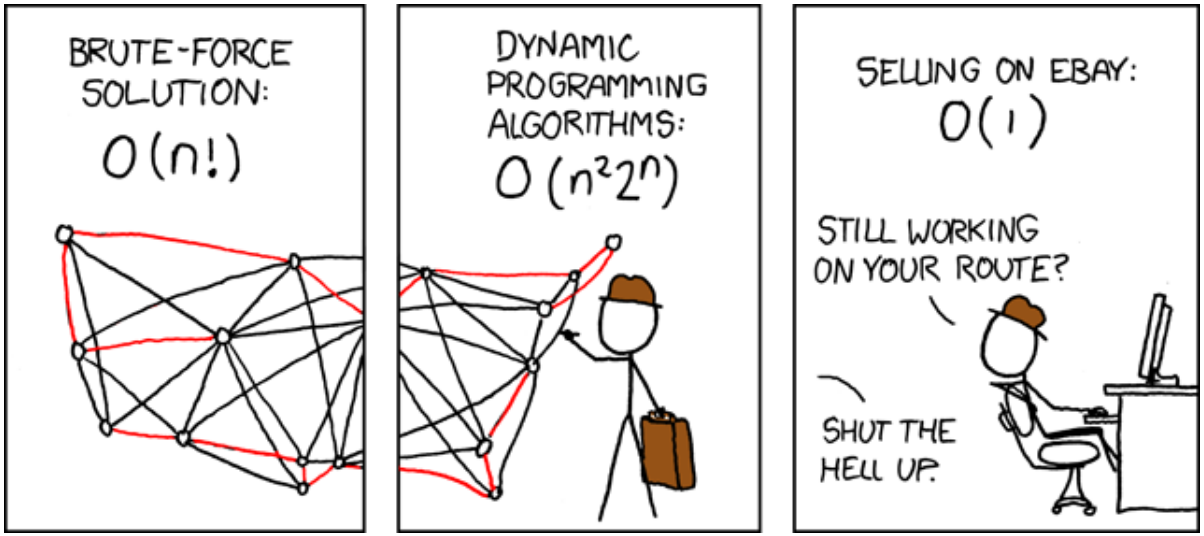
Is your solution correct?

- Try it on the sample input and a few examples
- Are there any EDGE CASES?
    - ICPC is *all or nothing* - one tiny edge case can cost you the whole problem
    - For minesweeper - edges of the board

Minesweeper - Solution

Think before you write any code!

Is your solution fast enough?



Minesweeper - Solution

Think before you write any code!

Is your solution fast enough?

N	Complexity	N	Complexity
10	O(n!), O(n^6)	100	O(n^4)
15	O(2^n * n^2)	1000	O(n^2)
20	O(2^n), O(n^5)	100K	O(nlogn)
50	O(n^4)	1M	O(n), O(logn), O(1)

Minesweeper - Solution

Think before you write any code!

How long will it take to write it?

- Teammates want to know that.
- Essential for team contests:
  - can prioritize actions/problems based on estimates
    - quick problems first
    - debugging vs implementation

#	AFFIL.	TEAM	SCORE		A	B	C	D	E	F	G	H	I
1		The Deepstackers	8	1288	1 (282 + 0)	1 (196 + 0)	0	2 (135 + 20)	2 (72 + 20)	0	1 (246 + 0)	0	2 (1 + 2
2		Geen Syntax	8	1410	2 (216 + 20)	1 (284 + 0)	0	2 (161 + 20)	1 (153 + 0)	0	1 (244 + 0)	0	2 (1 + 2
3	LMU	We don't have a team name	7	907	1	1 (183 + 0)	0	1 (141 + 0)	1 (66 + 0)	2	2 (255 + 20)	0	1 (9 0)
4		Rooftop Cornflakes	7	974	0	2 (185 + 20)	0	1 (54 + 0)	2 (41 + 20)	6 (297 + 100)	1	0	2 (9 20)
5		Netcraft-Bath	7	1108	2 (257 + 20)	1 (206 + 0)	0	3 (59 + 40)	2 (112 + 20)	0	0	0	1 (7 0)
6	KIT	Karlsruhe International Team	7	1193	1 (118 + 0)	3 (297 + 40)	0	2 (135 + 20)	1 (33 + 0)	0	0	0	4 (9 60)
7		Head of the River	7	1325	1 (134 + 0)	3	0	3 (217 + 40)	1 (69 + 0)	1 (273 + 0)	0	0	2 (1 + 2
8		Kompaktheit	7	1396	0	3 (263 + 40)	0	2 (84 + 20)	6 (90 + 100)	0	2 (234 + 20)	0	1 (1 + 0
9	KIT	false'); DROP TABLE teams; --	6	617	0	4	0	3 (97 + 40)	1 (24 + 0)	0	1 (197 + 0)	0	1 (7 0)
10		Tha Java guys and Emil	6	758	0	1 (165 + 0)	0	1 (100 + 0)	1 (122 + 0)	0	0	9	1 (4 0)
11		Lambdabamserne	6	796	0	2 (241 + 20)	0	2 (203 + 20)	1 (49 + 0)	0	0	5	1 (1 + 0
12	LEUVEN	King High	6	908	0	6 (290 + 100)	0	2 (110 + 20)	1 (126 + 0)	0	0	0	1 (1 + 0
13		cyberFAUbia	6	1034	1 (171 + 0)	3 (138 + 40)	0	4	1 (85 + 0)	0	0	0	1 (2 + 0
14		TriviaU	6	1108	4 (213	3	0	3 (111	1 (120	0	0	0	2 (1

Minesweeper - Solution

Think before you write any code!

How long will it take to write it?

- Teammates want to know that.
- Essential for team contests:
  - can prioritize actions/problems based on estimates
    - ICPC ranking: first by score, then by penalty
      - penalty = time submitted + wrong_submissions * 20

Minesweeper - Solution Revisited

Pretty easy:

1. Keep a count matrix
2. Loop through the board
   - if we find a mine (*) at i, j

     increment count for all non-mine adjacent fields
3. Print the count matrix

Correct? Yes

Fast? O(n^2) ⇒ Yes

Time? ~10 mins

Let's code it!

Java I/O

- Scanner - useful functions, slower

```
Scanner sc = new Scanner(System.in);
while (sc.hasNextInt()) {
  int n = sc.nextInt();
  // OR
  // double d = sc.nextDouble();
  // BigInteger bi = sc.nextBigInteger();
}
```

Java I/O

- BufferedReader - faster, no parsing functions

```
BufferedReader br =
    new BufferedReader(new InputStreamReader(System.in));
String line;
while ( (line = br.readLine()) != null) {
    int n = Integer.parseInt(line);
}
// useful function for parsing: line.split("delimiter")
```

Minesweeper - Solution

```
 for (int i = 0; i < n; i++) {
   char[] line = sc.nextLine().toCharArray();
   for (int j = 0; j < m; j++) {
     if ( line[j] != '*' ) continue;
     for (int l = i - 1; l <= i + 1  && l < n; l++)
       for (int c = j - 1; c <= j + 1 && c < m; c++)
         if (l >= 0 && c >= 0 && count[l][c] != -1)
            count[l][c]++;
       count[i][j] = -1;
   }
}
```

Minesweeper - Testing

Write tests for:

1. Any edge case
2. Very small inputs
3. Very large inputs, catch:
    a. Array out of bounds
    b. Overflow
4. Repeated test - makes sure you clear any intermediary data between test cases

Minesweeper - Testing

Generating large test cases - Python

```
import sys, random
MAX=100
print '{} {}'.format(MAX, MAX)
for i in range(MAX):
    for j in range(MAX):
        sys.stdout.write('.' if random.randint(0, 1) == 0 else '*')
    print ''
print '0 0'
```