A Bogus Induction Proof

P(n): for any x, $x^n = 1$.

1. [Basis] n=0. P(0) says that for all x, $x^0 = 1$ which is true.
2. The induction hypothesis is that P(k) holds for all $k \leq n$.
3. [Induction step] To prove P(n+1) note that

$$x^{n+1} = x^n \cdot x^n / x^{n-1}.$$

But by the induction hypothesis, $x^{n-1} = 1$ and $x^n = 1$

(that is, P(n) and P(n-1) are both true)

so $x^{n+1} = 1 \cdot 1 / 1 = 1$.

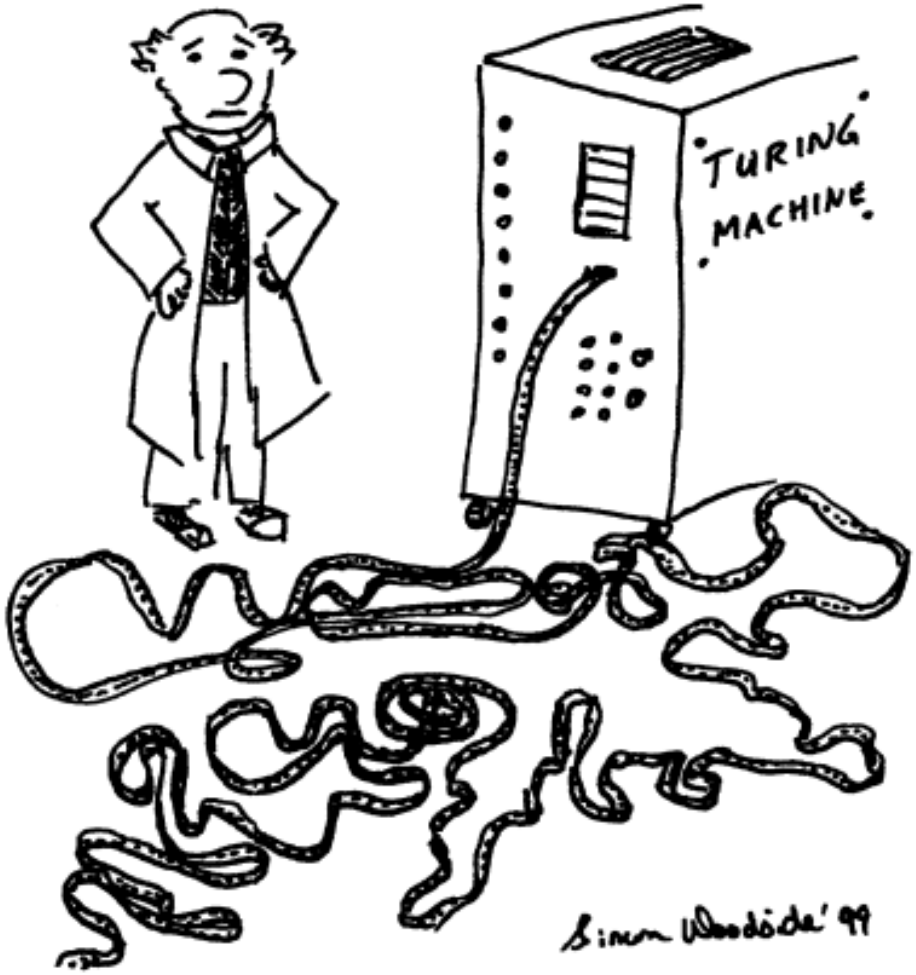
Where is the flaw in this argument?

CSC 320: Fall 2010

Dr. Wendy Myrvold

ECS 552

wendym@cs.uvic.ca



Turing solves the halting problem, only to discover that the REAL problem with his machine is what to do with all the tape.

Cartoon by Simon Woodside: simonwoodside.com/weblog/images/2005/turing.gif

Announcements

- Assignment 1 is posted- due Fri. May 25.
- Tutorial 1 is posted for Tues. May 15.
  You will benefit more if you do the questions in advance.
- A midterm study aid has a reading list, for now read Chapter 1.
- Friday- bring schedule to class to help me in selecting office hours.
- Make sure you sign the attendance sheet every class to get credit for attending.
- Powerpoint slides will be posted.

Important Announcements from Jane Guy

CSC 320 Prerequisites: CSC 225,
MATH 222 or PHYS 242.
If you do not have the prerequisites you should drop the class now to avoid paying for it.

If you are repeating a course for the 3rd or even the 4th time you must ask for permission to do so.
A form to apply for permission is available at the Computer Science Advising Centre ECS 514.

Questions: See Jane Guy, ECS 512

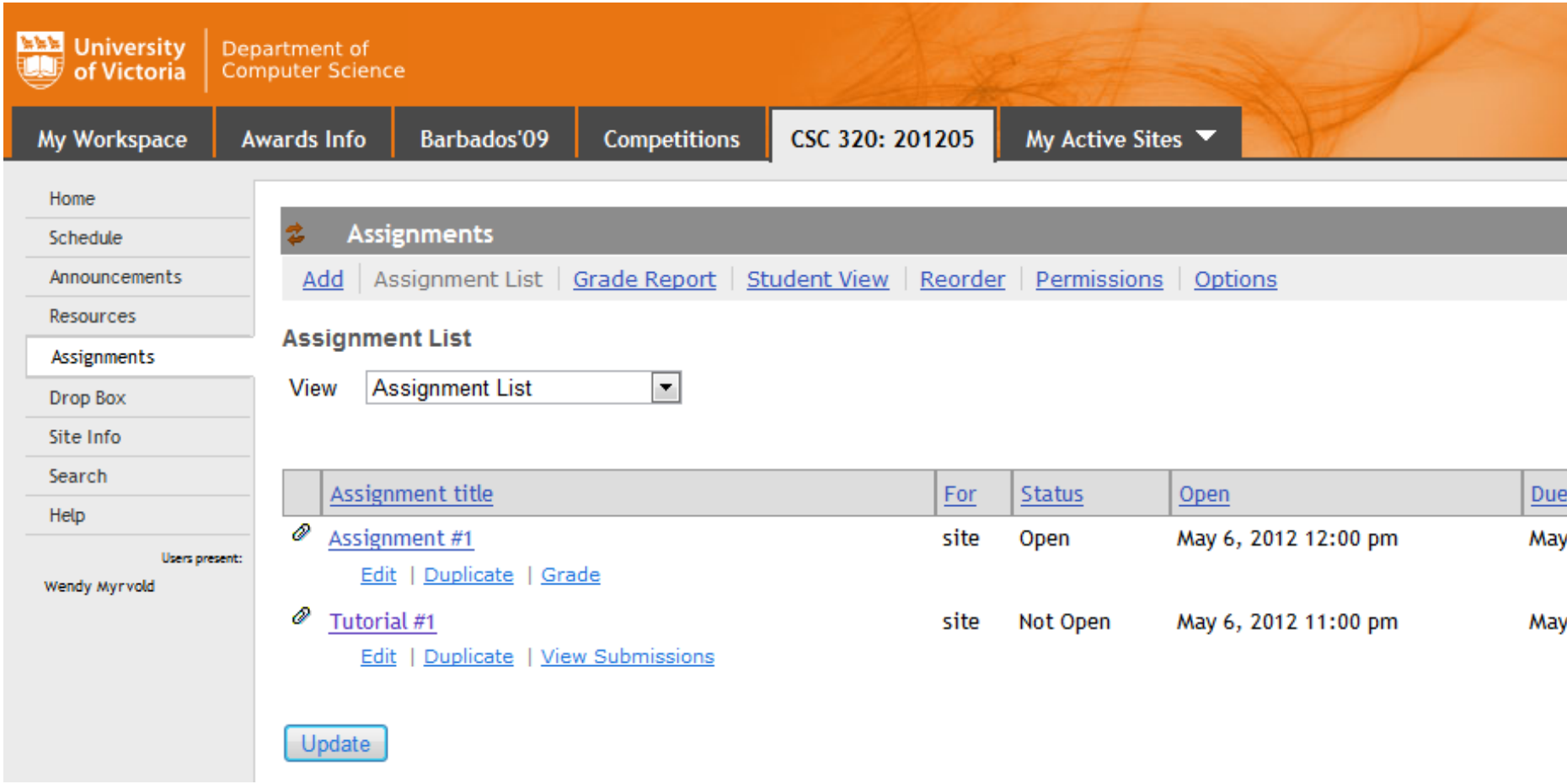
Class Materials and Announcements

Connex will be used for posting model solutions and other private class resources and also for sending e-mail announcements to the class.

If you have connex? next to your name on the class signature list, this means I did not see your name on Sunday on the class roster.

Instructions for getting connex access are at the bottom of our class web page.

Assignment #1 and Tutorial #1 are on Connex:



But also available from class web pages.

Outline for Lecture 1

- Who is the instructor?
- My research interests
- Logistics for CSC 320- the critical points are included on the course outline
- Brief overview of course content- don't worry about taking notes today

by Mark A. Hicks, illustrator.

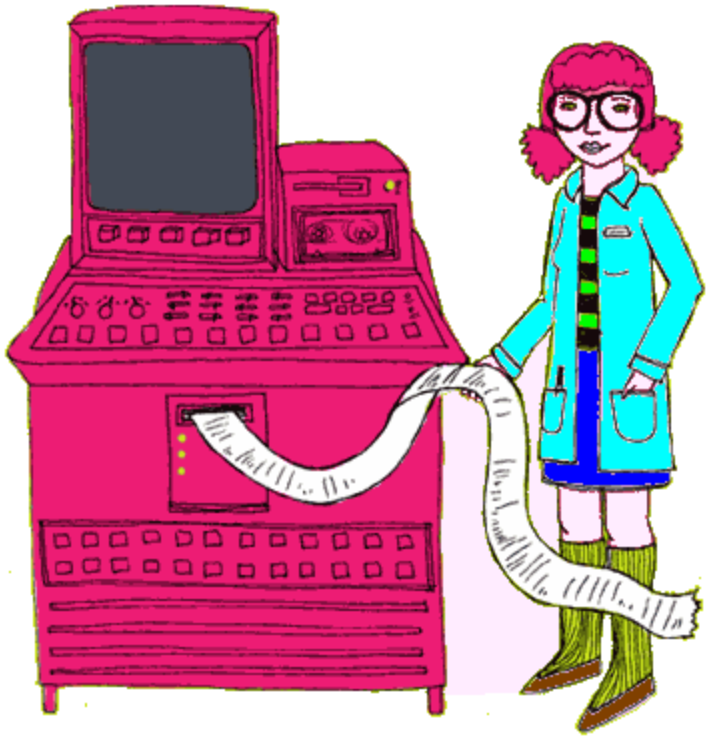
About me:

M.Sc. : Computer Science, McGill University, 1983

M.Math. : Combinatorics and Optimization, University of Waterloo, 1984

Ph.D. in Computer Science: Waterloo, 1988

University of Victoria: started in 1988, currently a full professor



From: Gurl Guide to programming.

Jennifer

Sean

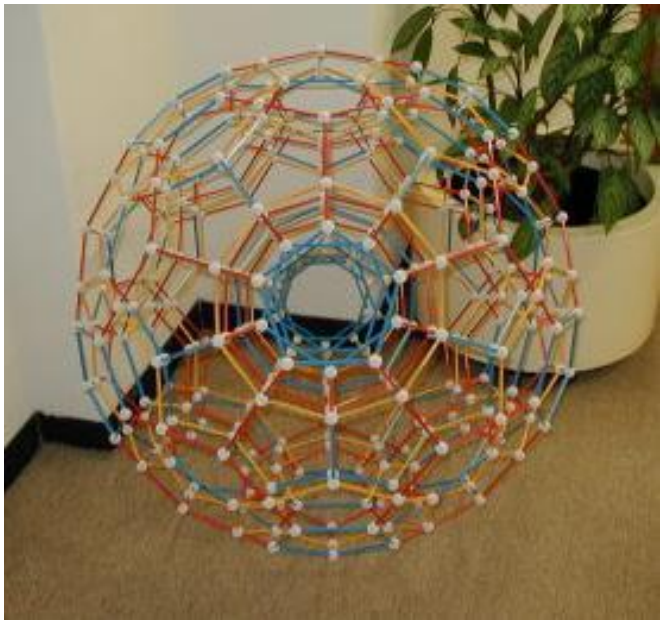
Bring your parents to work day at Google.

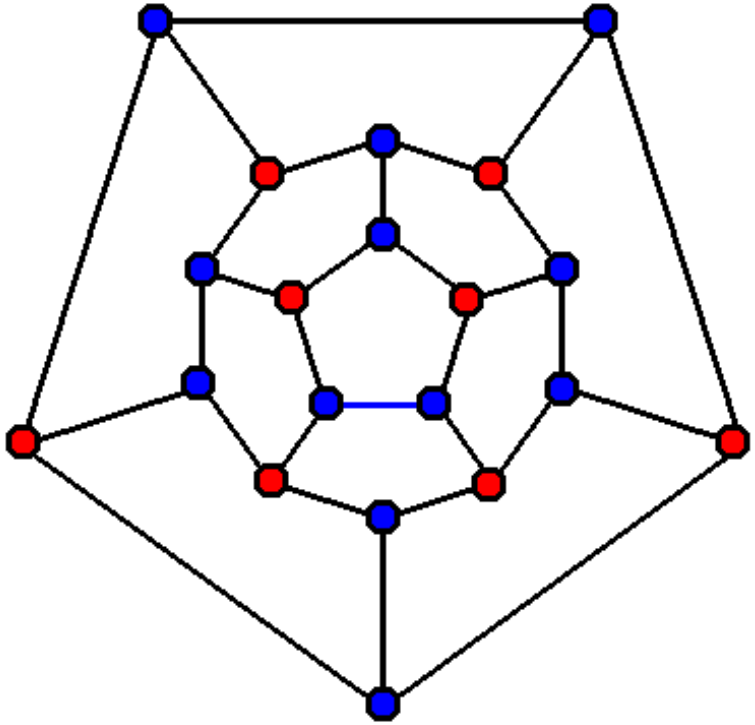

M
A
CCM

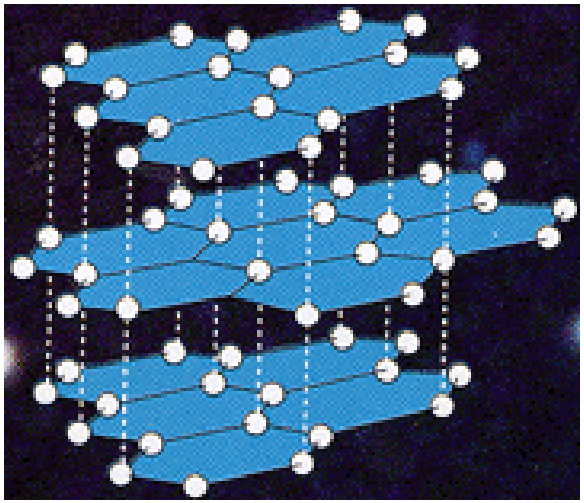
My Research: Large Combinatorial Searches

Independent Set:

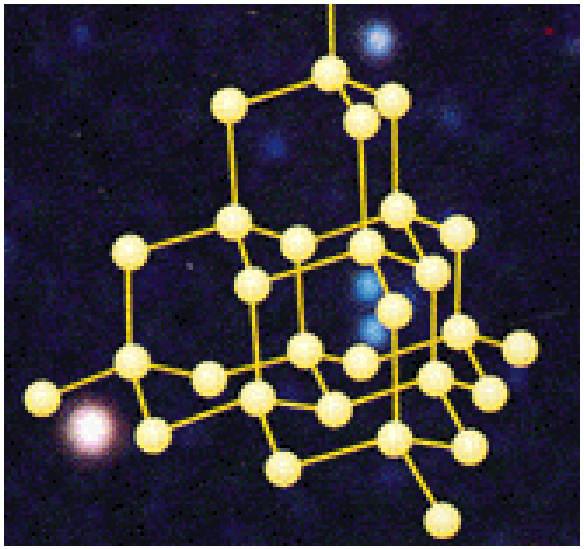
Set of vertices which are pairwise non-adjacent







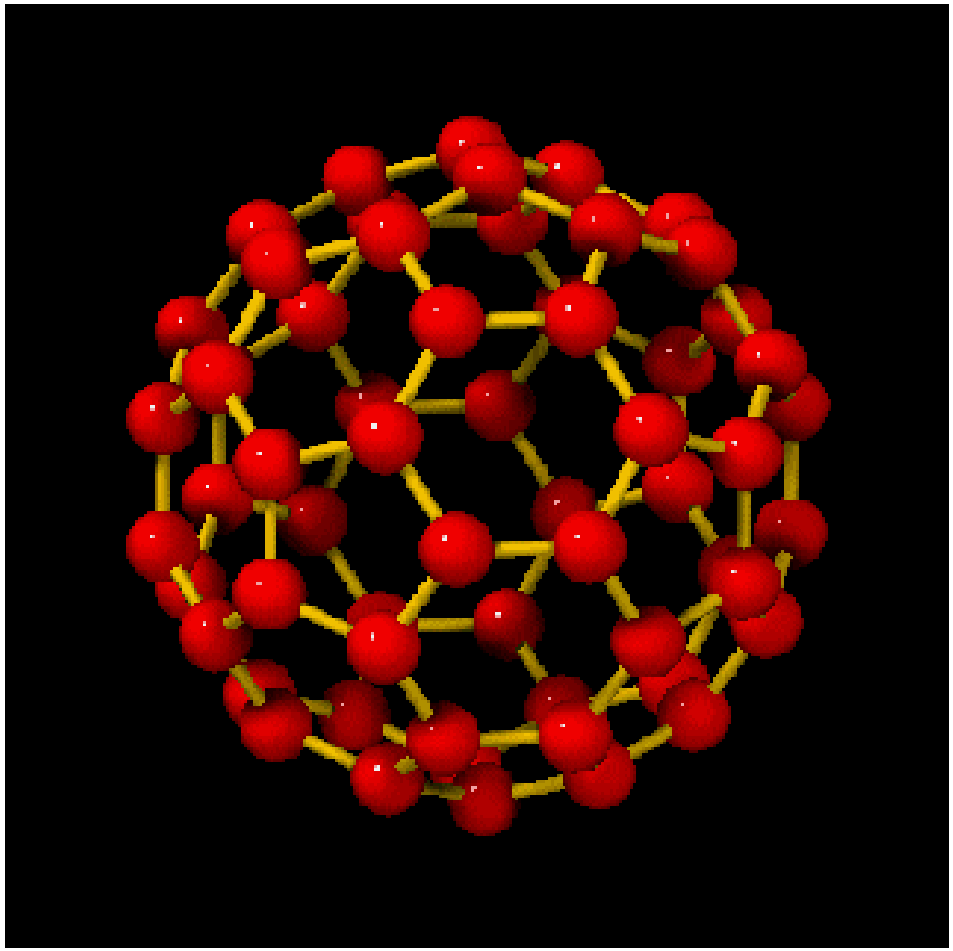
Graphite



Diamond

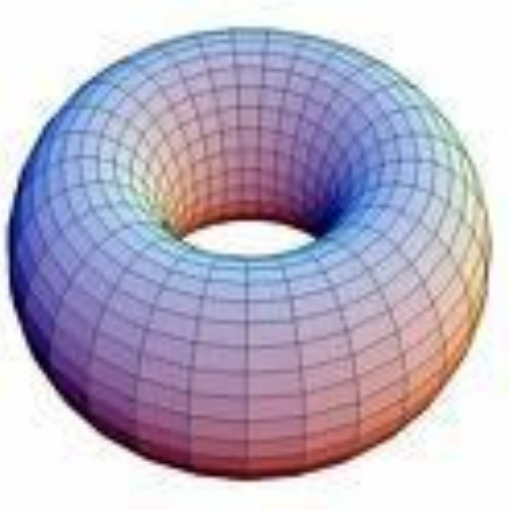
Fullerenes:

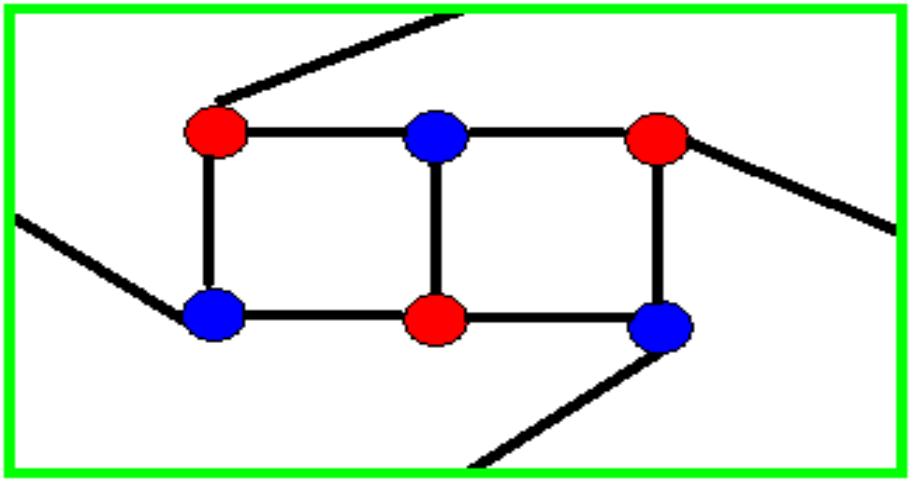
Working with Patrick Fowler (chemist)

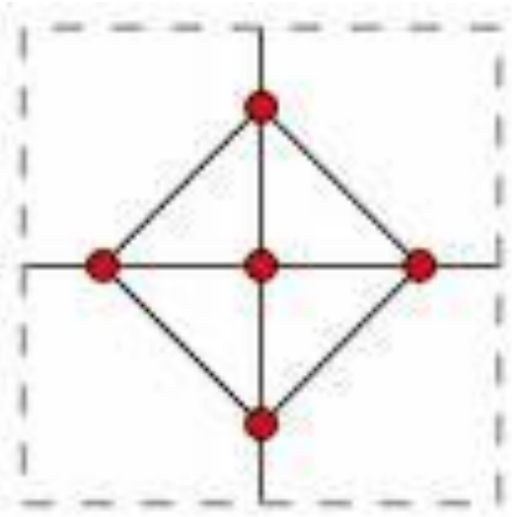


Topological Graph Theory:

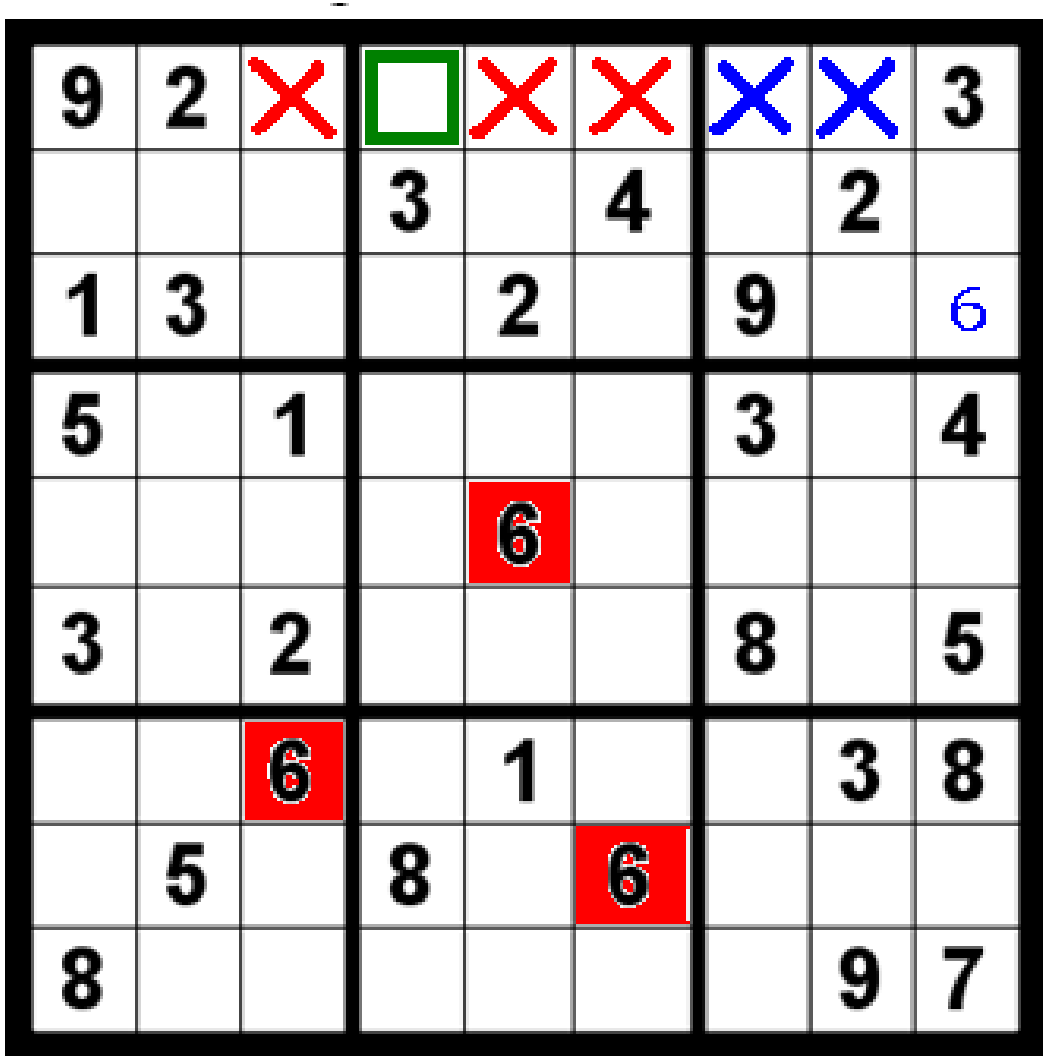
Algorithms and Obstructions



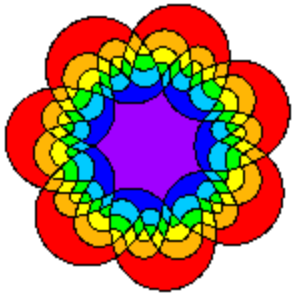




Latin Squares

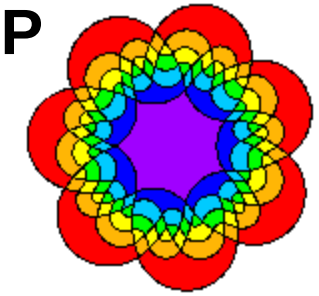


Please come talk to me if you are looking for Honours project research topics or for an NSERC undergraduate research project.



COMBINATORIAL ALGORITHMS GROUP
University of Victoria

http://www.cs.uvic.ca/~wendym/cag



Our research interests include:

Graph Theory and Graph Algorithms
Combinatorics
Combinatorial Algorithms
Computational Geometry
Randomized Algorithms
Computational Complexity
Network Reliability
Topological Graph Theory
Computational Biology
Cryptography
Design Theory

Join our listserv to get information about conferences and research talks.

Undergrads are welcome to all events.

CSC 320 Logistics

Course Website: http://www.cs.uvic.ca/~wendym/320.html

Instructor: Dr. Wendy Myrvold
Email: wendym@csc.uvic.ca

I answer all student e-mails. If you do not get a response in a reasonable time frame please find out why the e-mail did not work.

Office: ECS 552
Phone Number: 472-5783 (use e-mail for a faster response)
Office Hours: See course web page. This week: WF 11:30-1:30 or by apt.
Please tell me if you plan to come by.

Lecture Schedule: TWF 10:30 - 11:20 p.m. SSM A 104

Tutorial Schedule: T 3:30 - 4:20pm ECS 108

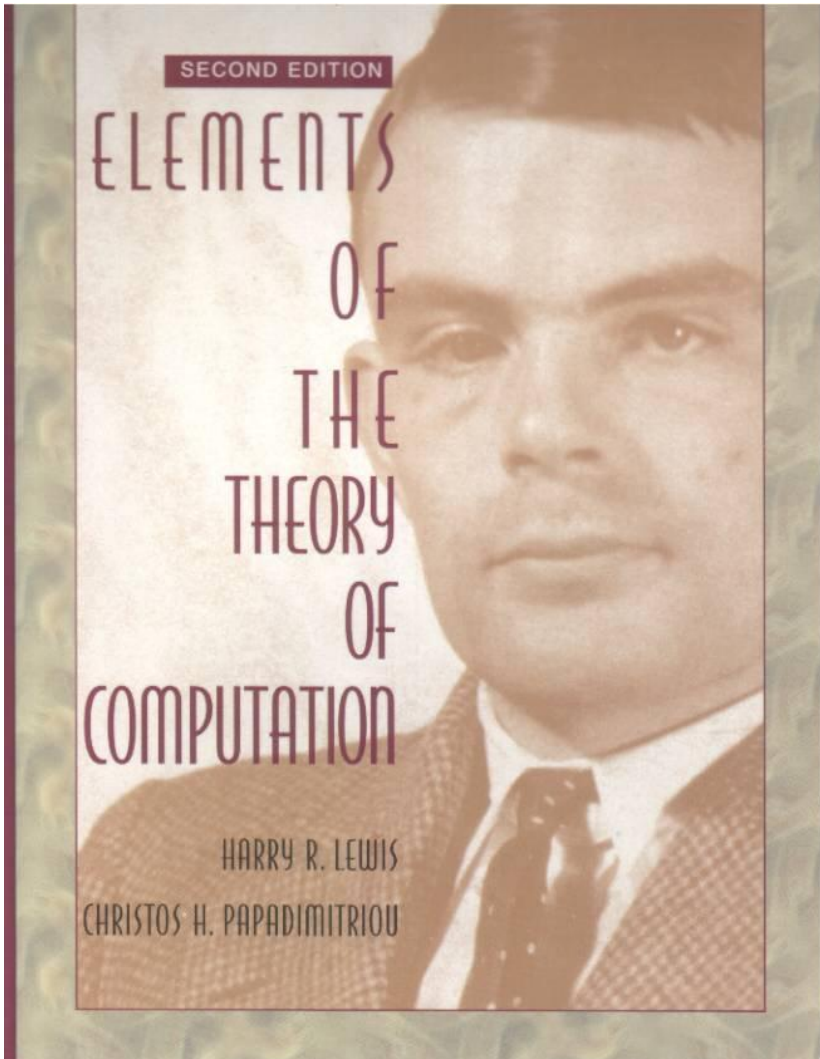
Starts May 15.

Course Textbook

Elements of the Theory of Computation, 2nd Edition by Harry R. Lewis and Christos H. Papadimitriou, Prentice-Hall, 1998

See the midterm study aid for a reading list for the first half of the course.

Selected course notes will also be posted online.



CSC 320 Grading

You collect points from assignments, the midterm and participation:

ITEM	REQUIRED FOR FINAL	MAX POINTS
Assignments-5	10/20 = 50%	20
Midterm Wed. June 27	10/25 = 40%	25
Participation	2.5/5 = 50%	5

Computation of Final Grade

If you get less than 50% on the assignments 40% on the midterm or 50% of the participation marks, you cannot write the final- your grade is N.

Otherwise:

You have Q points where $22.5 \leq Q \leq 50$.

Your final numeric grade is:

Q plus your score on the final exam

weighted so that it is out of (100 - Q).

How to get an A+ in CSC 320

With 23 points:

A+	87
A	80.5
A-	74
B+	70
B	65
B-	61
C+	54.5
C	48
D	35

With 50 points:

A+	80
A	70
A-	60
B+	54
B	46
B-	40
C+	30
C	20
D	0

Keys to Success

Attend all classes and tutorials.

Do all your homework.

Come see me (early and often) if you need help. I love working with students. Ask questions in class as well.

Join a study group but prepare your final submissions independently.

Work old midterms and final exams as practice for your midterm and final.

Don't be afraid of generating incorrect solutions- real mathematicians make many mistakes in the process of creating new mathematics.

Students with a disability

Please let me know as soon as possible how I can accommodate your disability.

It's sometimes possible to go beyond what is first offered by the disability center.

CSC320: Theory of Computation

Computation: processing of information based on a finite set of operations or rules.

- paper and pencil arithmetic
- abacus
- calculator
- digital computers
- programs in C/Java
- cells/DNA?
- human brain?
- quantum computers?

Desirable properties of a theory

Generality

- technology independent
- ignores inessential details

Precision

- formal mathematical model
- able to prove things about what can and cannot be computed

Representing Data

Alphabet: finite set of symbols

Ex. { a, b, c, … , z}

Strings: finite sequence of alphabet symbols

Ex. abaab, hello, cccc

Inputs and outputs of computations:

represented by strings.

ε represents an empty string (length 0)

Examples of Problems

Given a string x, does x have an even number of a's?

Given a string of symbols, does it represent a syntactically correct C program?

Given an integer p, is p prime?

Given a JAVA program, are there any inputs for which it gets stuck in an infinite loop?

Language: set of strings

First names of students taking CSC 320:

{Adrian, Alexander, Amanda, Braden, Brandon, Brodie, Catlin, Christopher, Daniel, Ding, Dong, Erik, Ethan, Evian, Gordon, Grigory, Hao, Heng, Jonathan, Jordan, Lin, Matt, Matthew, Mauricio, Meghan, Michael, Ramtin, Richard, Rui, Scott, Wenchong, Xiuxia, Xiyu, Zhuoli}

Strings over {a,b} with even length:

{ε , aa, ab, ba, bb, aaaa, aaab, aaba, …}

Syntactically correct JAVA programs.

How hard is it given a language L and a string w to answer the question: Is w in L? yes or no?

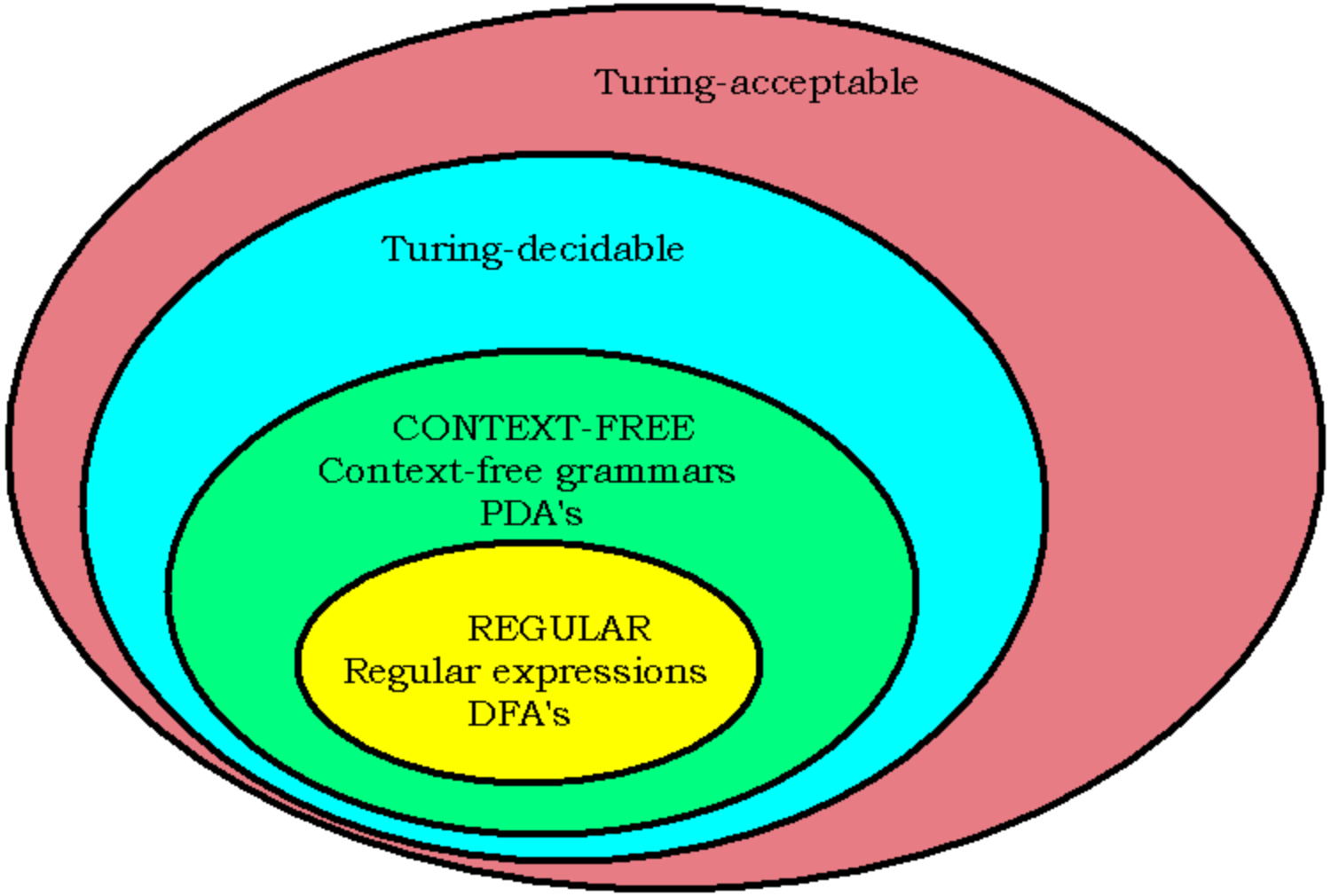
Regular languages: finding patterns in strings

Context-free languages: compiler design, parsing computer languages such as C, JAVA, HTML

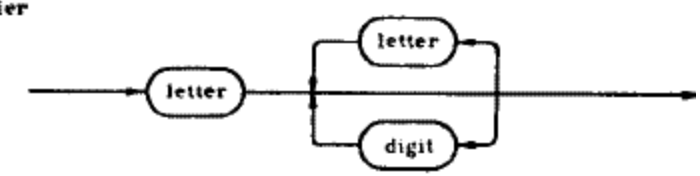
Turing-decidable languages: yes/no questions which are computable on a computer

Turing-acceptable languages: an algorithm can be designed which halts when the answer is yes but possibly computes forever when the answer is no.

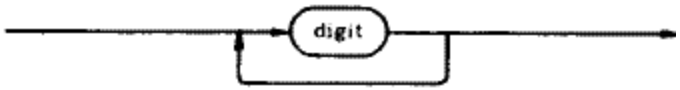
Classes of Languages



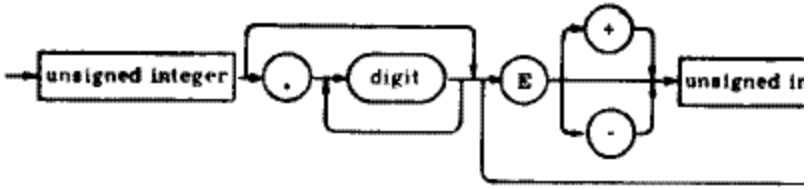
identifier



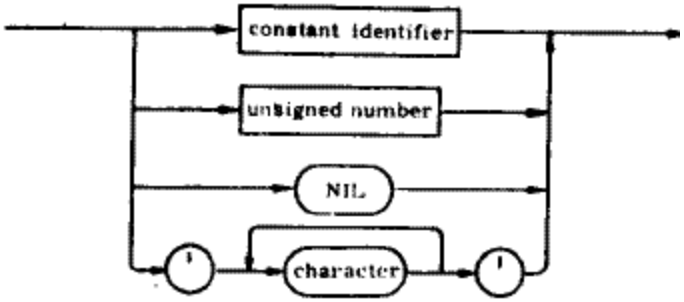
unsigned integer



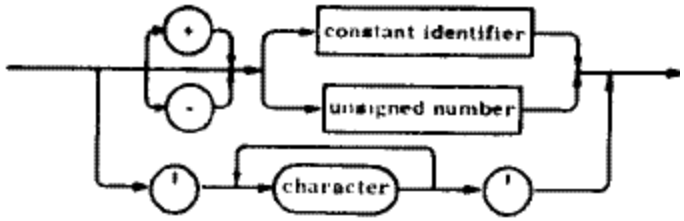
unsigned number



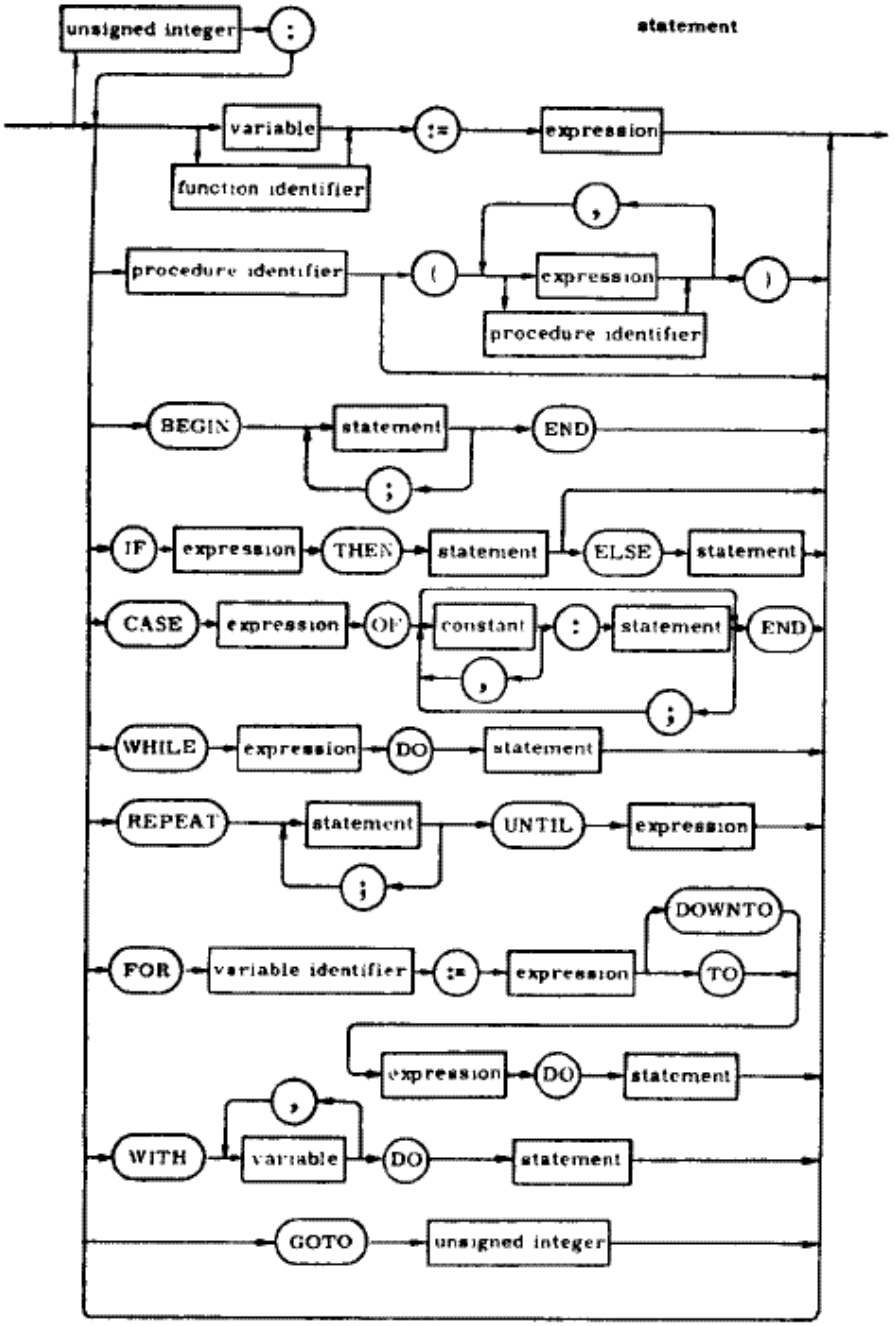
unsigned constant



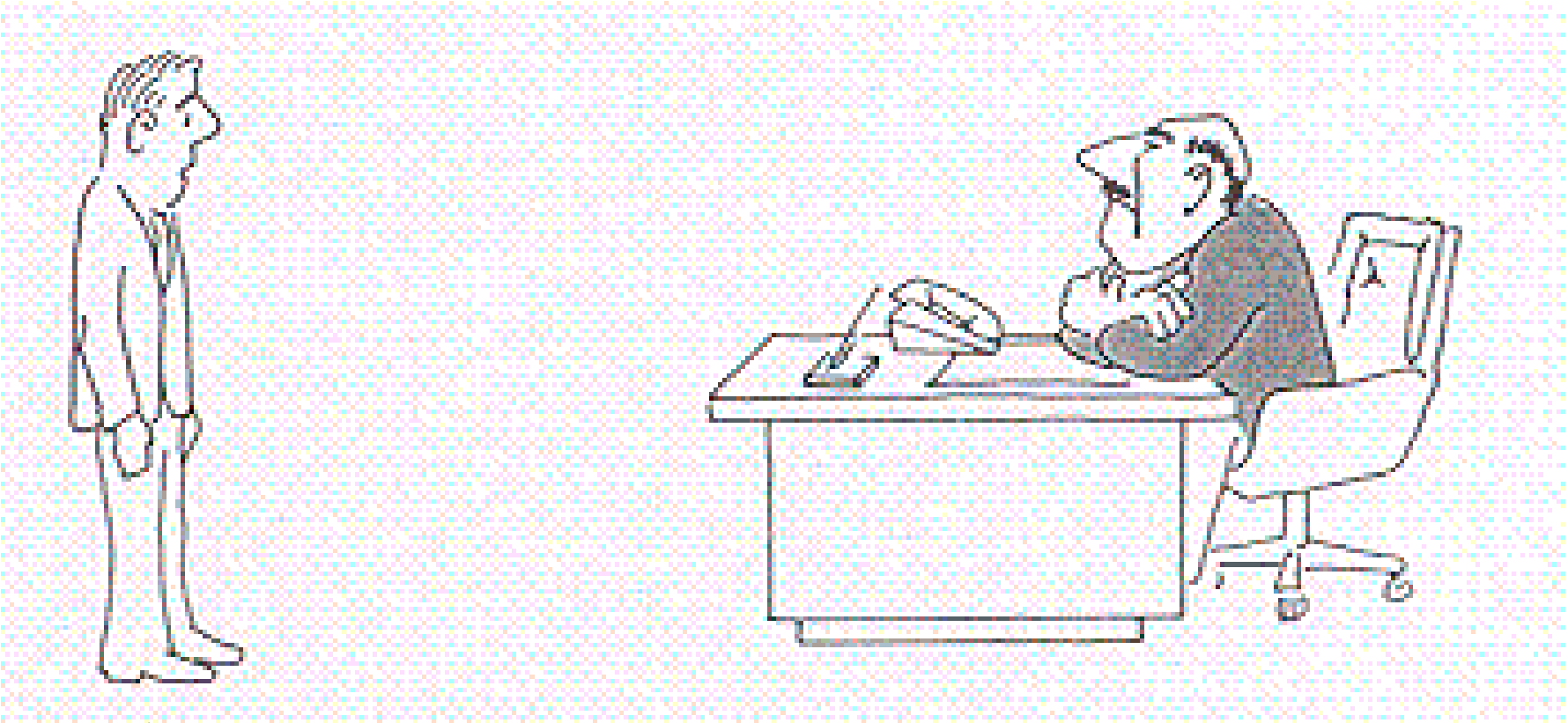
constant



statement



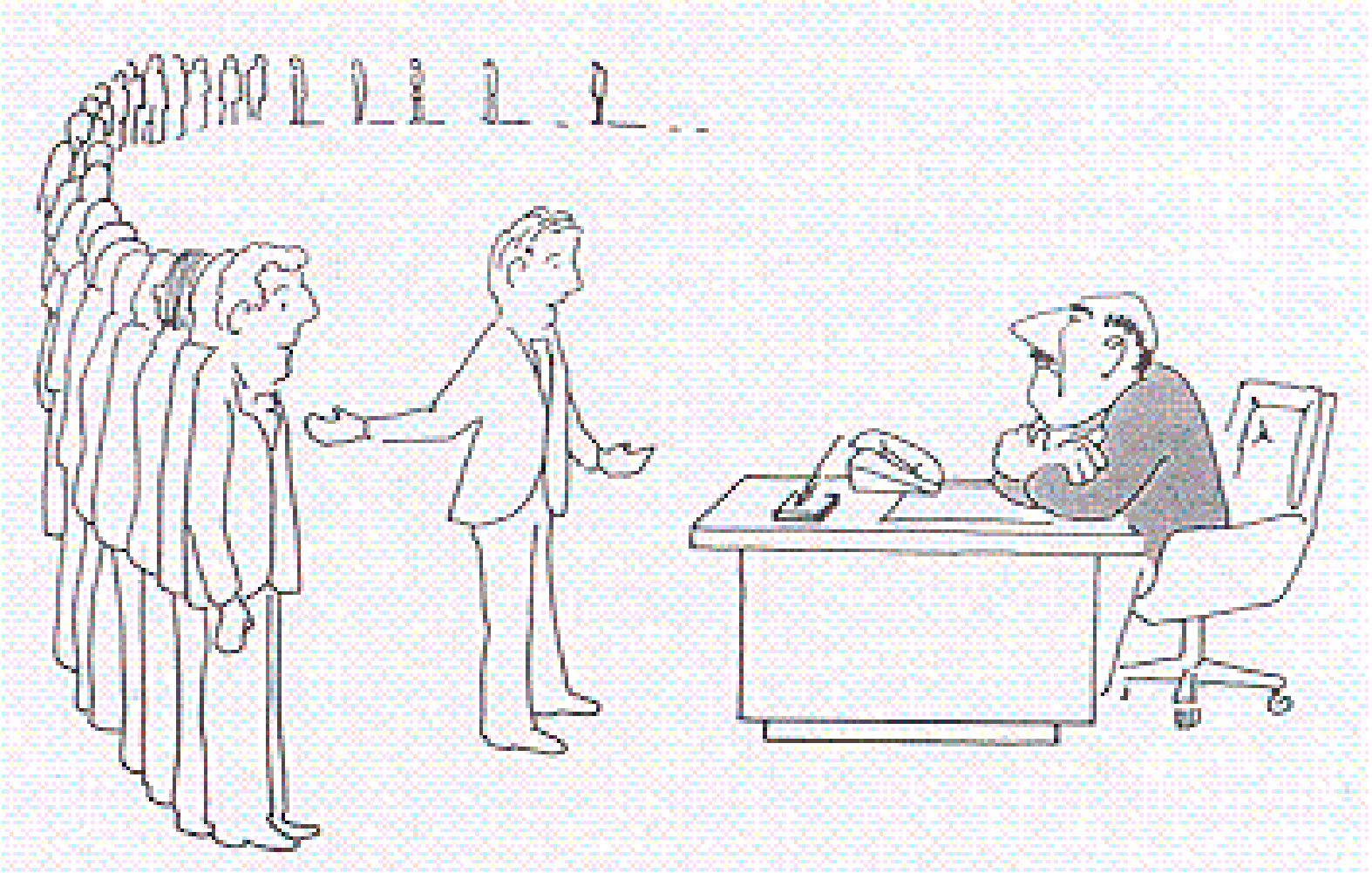
NP-completeness



I can't find an efficient algorithm,

I guess I'm just too dumb.

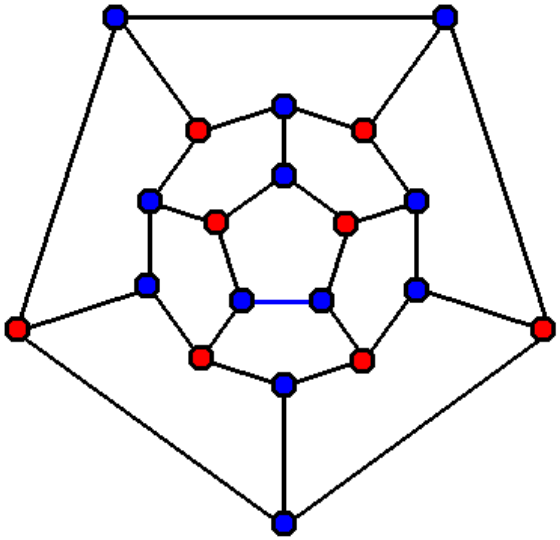
I can't find an efficient algorithm, because no such algorithm is possible.



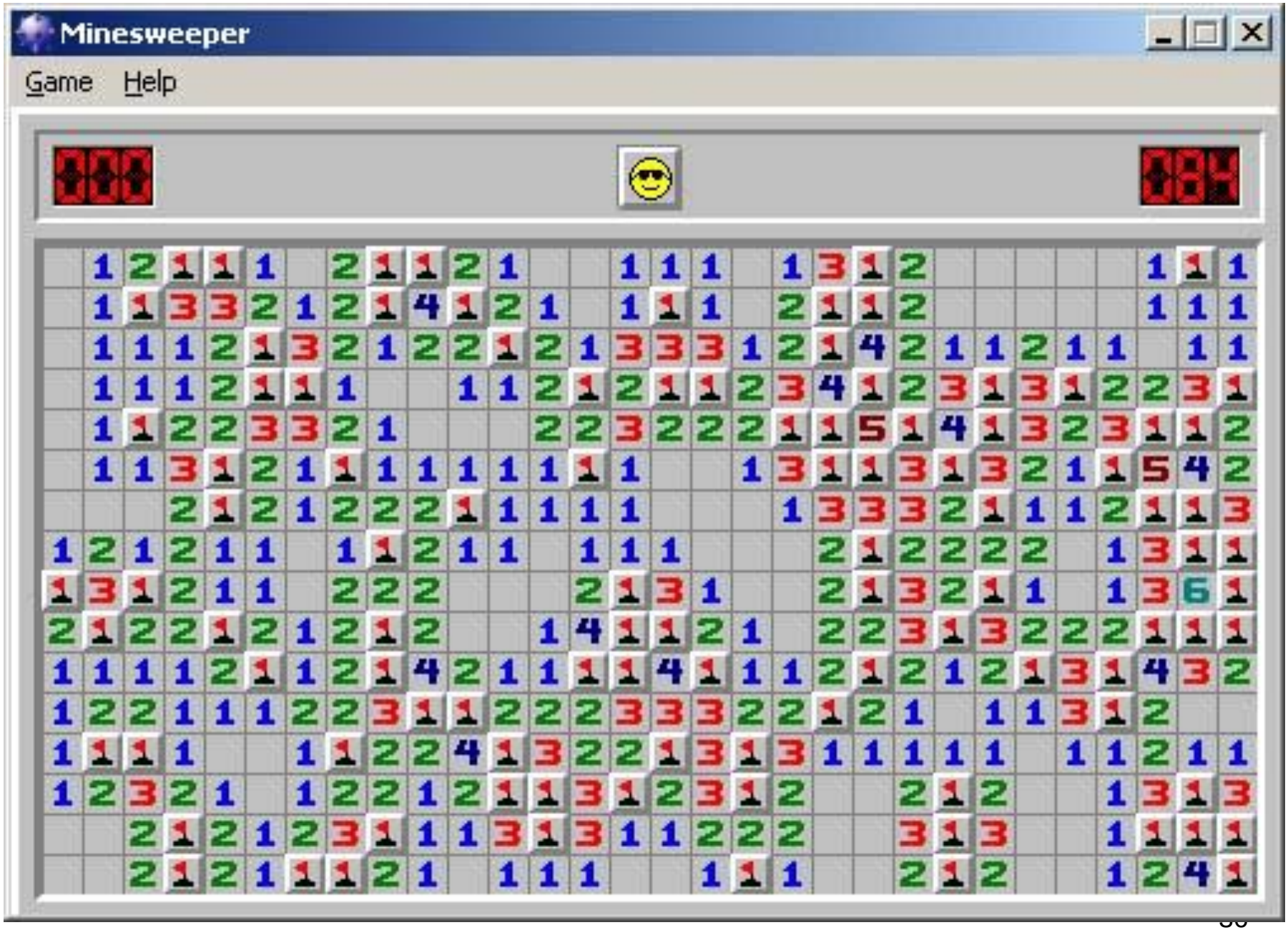
I can't find an efficient algorithm, but neither can all these famous people.

Some NP-complete Problems

- Graph 3-colouring
- Travelling Salesman Problem
- Independent Set
- Boolean Satisfiability
- Bin packing
- Scheduling



Either all of these of none of them has a polynomial time ($O(n^k)$ for a constant k) algorithm. There is a million dollars for the first person to prove this either way.


Minesweeper
Game
Help

Ian Stewart on Minesweeper:

It's not often you can win a million dollars by analysing a computer game, but by a curious conjunction of fate, there's a chance that you might. However, you'll only pick up the loot if all the experts are wrong and a problem that they think is extraordinarily hard turns out to be easy.

The prize is one of seven now on offer from the newly founded Clay Mathematics Institute in Cambridge MA, set up by businessman Landon T. Clay to promote the growth and spread of mathematical knowledge, each bearing a million-buck price-tag. The computer game is Minesweeper, which is included in Microsoft's Windows operating system, and involves locating hidden mines on a grid by making guesses about where they are located and using clues provided by the computer. And the problem is one of the most notorious open questions in mathematics, which rejoices in the name 'P=NP?'.

You won't win the prize by winning the game. To win the prize, you will have to find a really slick method to answer questions about Minesweeper when it's played on gigantic grids and all the evidence suggests that there isn't a slick method. In fact, if you can prove that there isn't one, you can win the prize that way too.

Transferable Skills

- Formal specification of problems.
- Ability to identify correct and incorrect solutions and justify your answers.
- Enhanced ability to read and write proofs.
- Background needed for compiler design.
- Appreciation of what can and cannot be computed using a computer.