Proof of the Day:

Prove by induction that the number of binary strings of length k is 2^k.

For example:

The binary strings of length 3 are:

000, 001, 010, 011,
100, 101, 110, 111

and there is 8 of them, and $8 = 2^3$.

Prove by induction that the number of binary strings of length k is 2^k.

For example:

The binary strings of length 3 are:

000, 001, 010, 011,
100, 101, 110, 111

and there is 8 of them, and $8 = 2^3$.

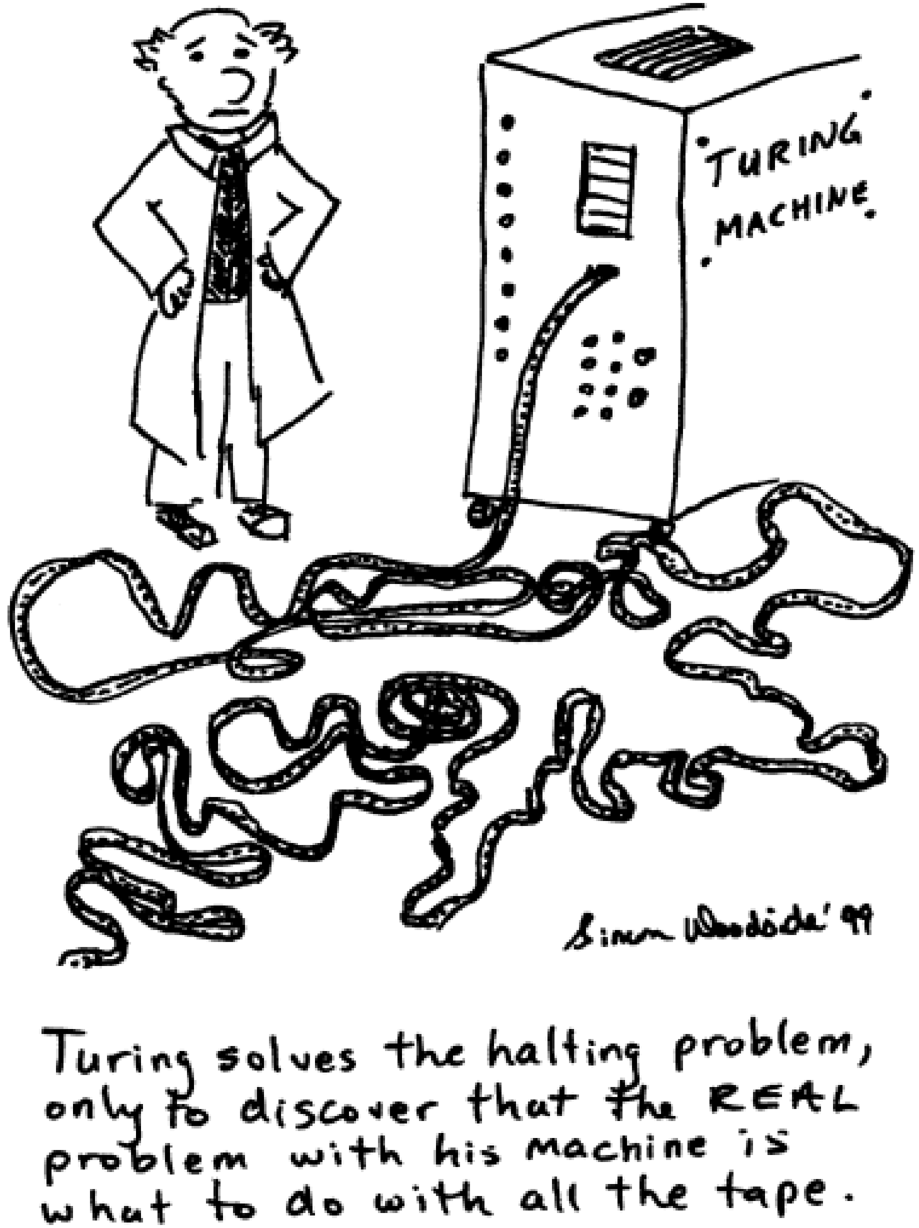
CSC 320:

Summer 2017

Dr. Wendy Myrvold

ECS 552

wendy@uvic.ca



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

Announcements

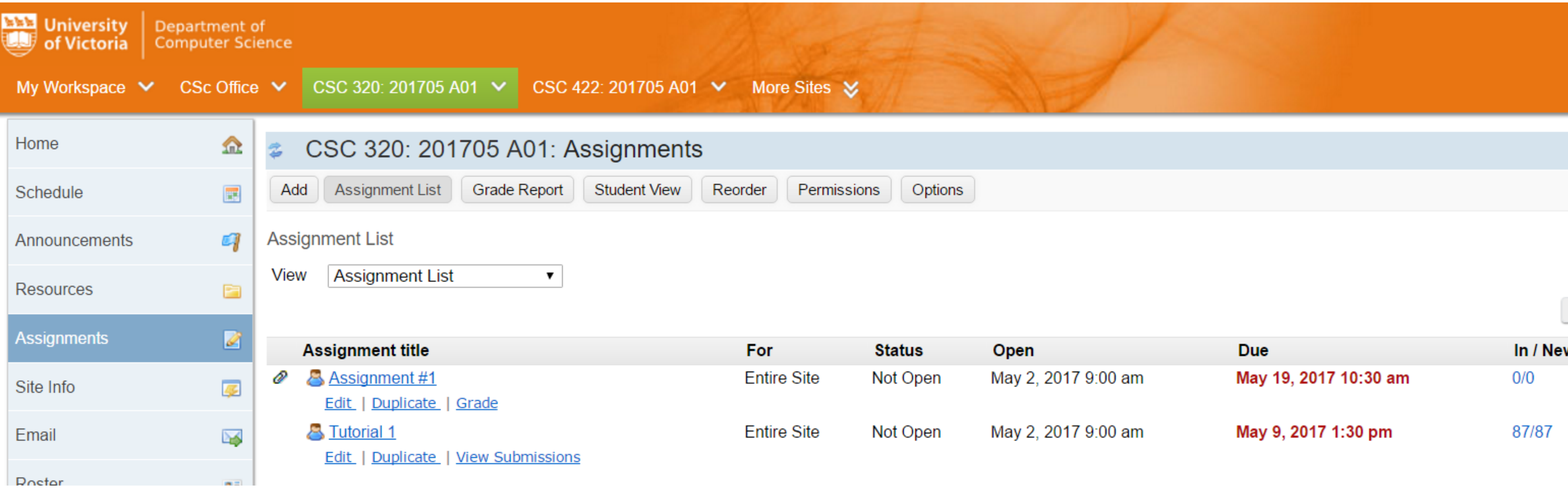
- Assignment 1 is posted- due Fri. May 19.
  Submissions: on paper in class, code must also be uploaded to connex.
- Tutorial 1 is posted for Tues. May 9.
  You will benefit more if you do the questions in advance.
- A midterm study aid has a reading list, for now read Chapter 1.
- Powerpoint slides will be posted: click on the "Selected class notes" link on the course web page.

Class Materials and Announcements

Connex: calendar, electronic assignment submission (for programs), links to assignments and tutorials on class web pages, model solutions and other private class resources, sending e-mail announcements to the class.

Course web pages: office hours, assignments, tutorials, projected schedules, class notes, old exams, study aids. No password required to access, accessible when connex is down.

Assignment #1 and Tutorial #1 are on Connex:



But also available from class web pages.

You can upload your programs at the bottom of the assignment page. Submit 2 files:
1. gen.c or Gen.c   2. new_ham.c or New_Hamilton.java

Your programs should be uploaded here:

gen.c or Gen.java

new_ham.c or New_Hamilton.java

Additional resources for assignment

http://webhome.cs.uvic.ca/~wendym/courses/320/17/a1_17.html (1 KB; May 1, 2017 1:40 pm)

Submission
This assignment allows submissions by attaching documents only. Use the Add Attachments button below to attach 1 or more documents.

Attachments

No attachments yet

Add Attachments

Honor Pledge: I have neither given nor received aid on this assignment.
(You must respond to submit your assignment.)

Back to list

You can upload files as many times as you like until the deadline. Please delete old versions.

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

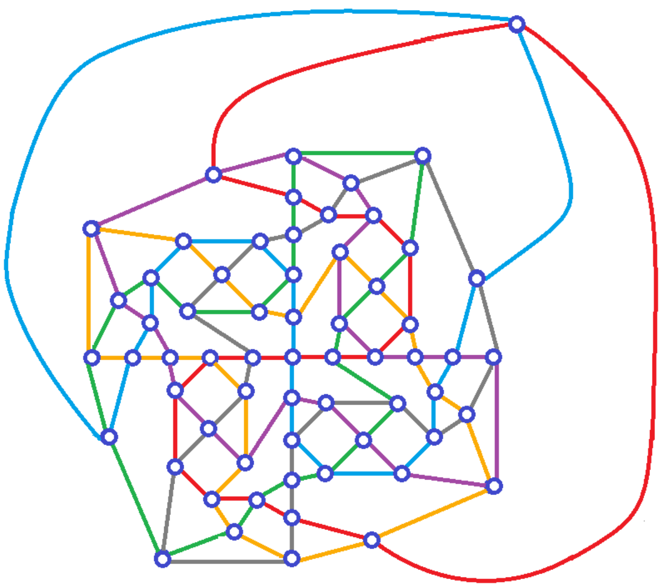
About me:

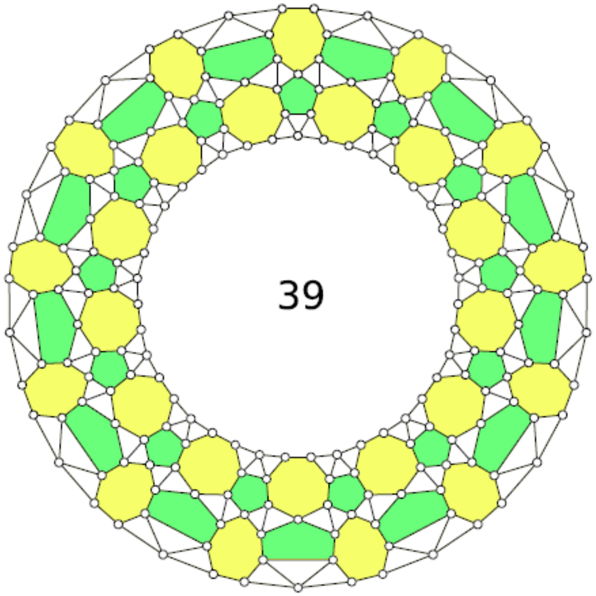
B.Sc. : Computer Science, McGill University, 1983

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





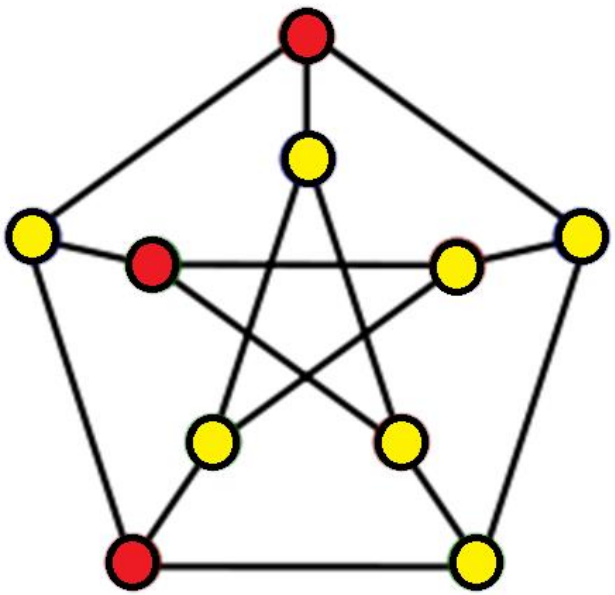
Bring your parents to work day at Google.

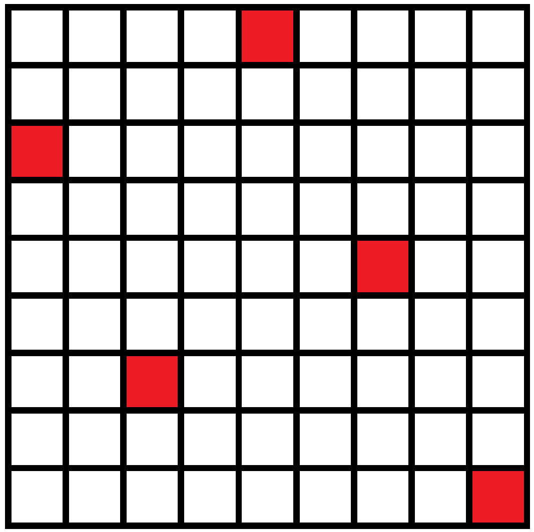

CCM

My Research: Large Combinatorial Searches

Dominating set S:

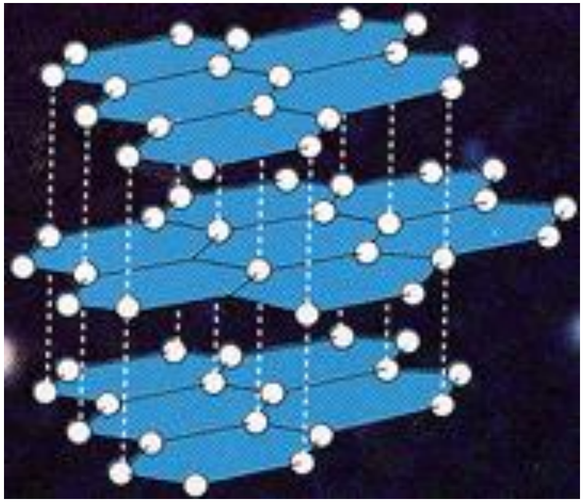
Every vertex is in S or is adjacent to a vertex in S.



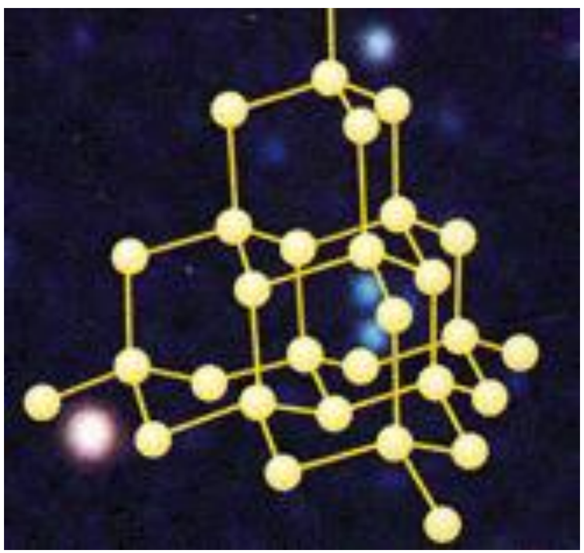


Fullerenes:

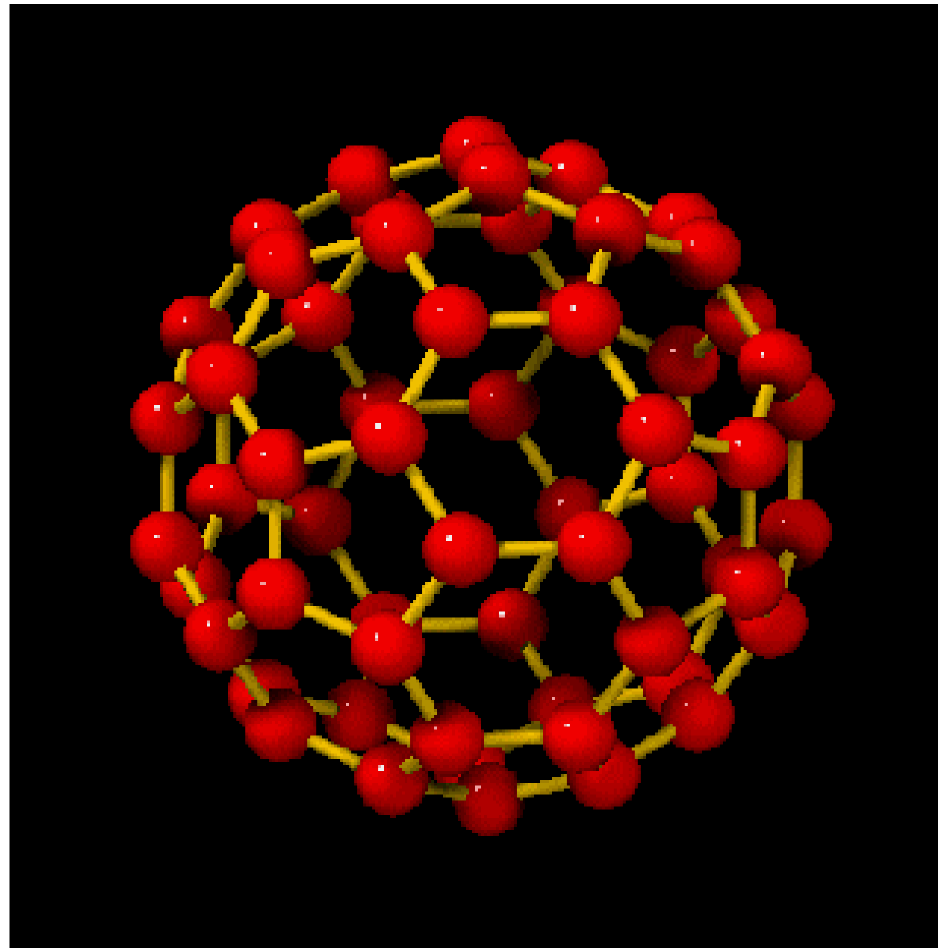
Working with Patrick Fowler (chemist)



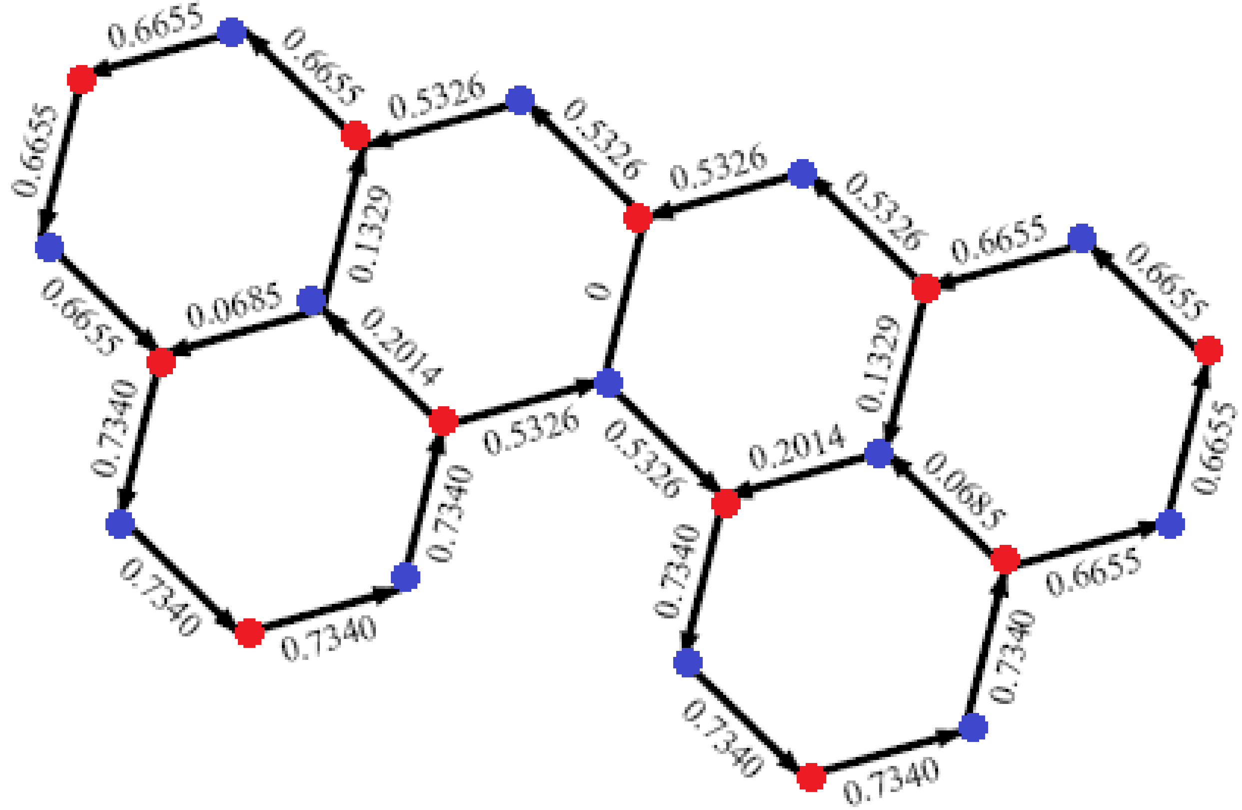
Graphite



Diamond

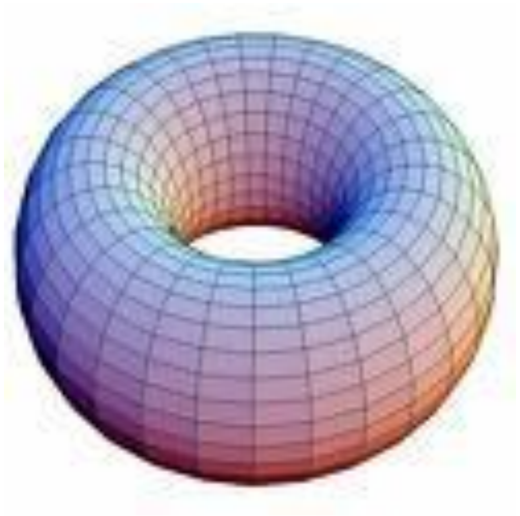


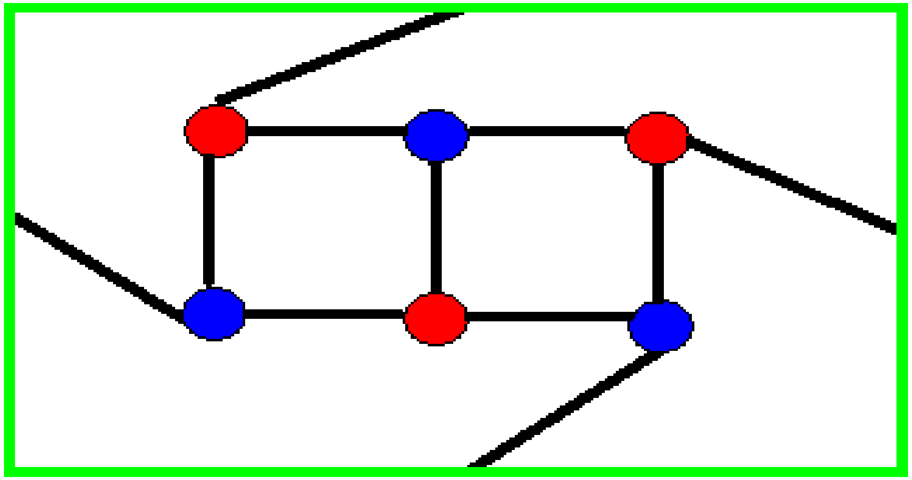
Currents in benzenoids (joint work with Patrick Fowler, a chemist):

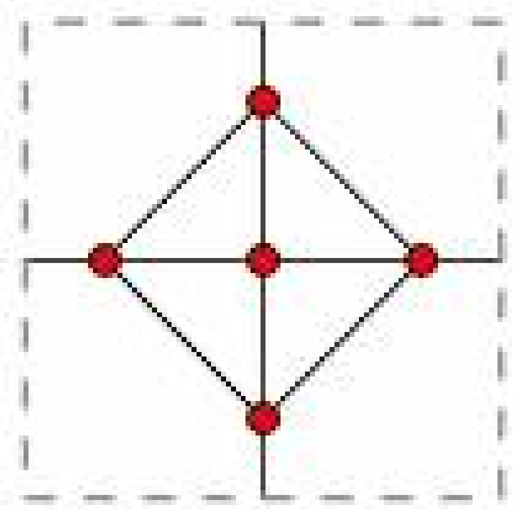


Topological Graph Theory:

Algorithms and Obstructions







Latin Squares

9	2							3
			3		4		2	
1	3			2		9		6
5		1				3		4
				6				
3		2				8		5
		6		1			3	8
	5		8		6			
8							9	7

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

CSC 320 Logistics

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

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

Put "CSC 320: meaningful subject header" in your e-mail messages.

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

Office hours: TWF 11:30, TF 4:00 until all questions answered.

Tutorial (T01): T 1:30 - 2:20 p.m. Cle A 307
Tutorial (T02): T 2:30 - 3:20 p.m. Cle A 307

Starts May 9

See course web page for tutorial schedule.

The first four are:

Tutorial #1: Tues. May 9.
Tutorial #2: Tues. May 16.
Tutorial #3: Tues. May 30.
Tutorial #4: Tues. June 13.

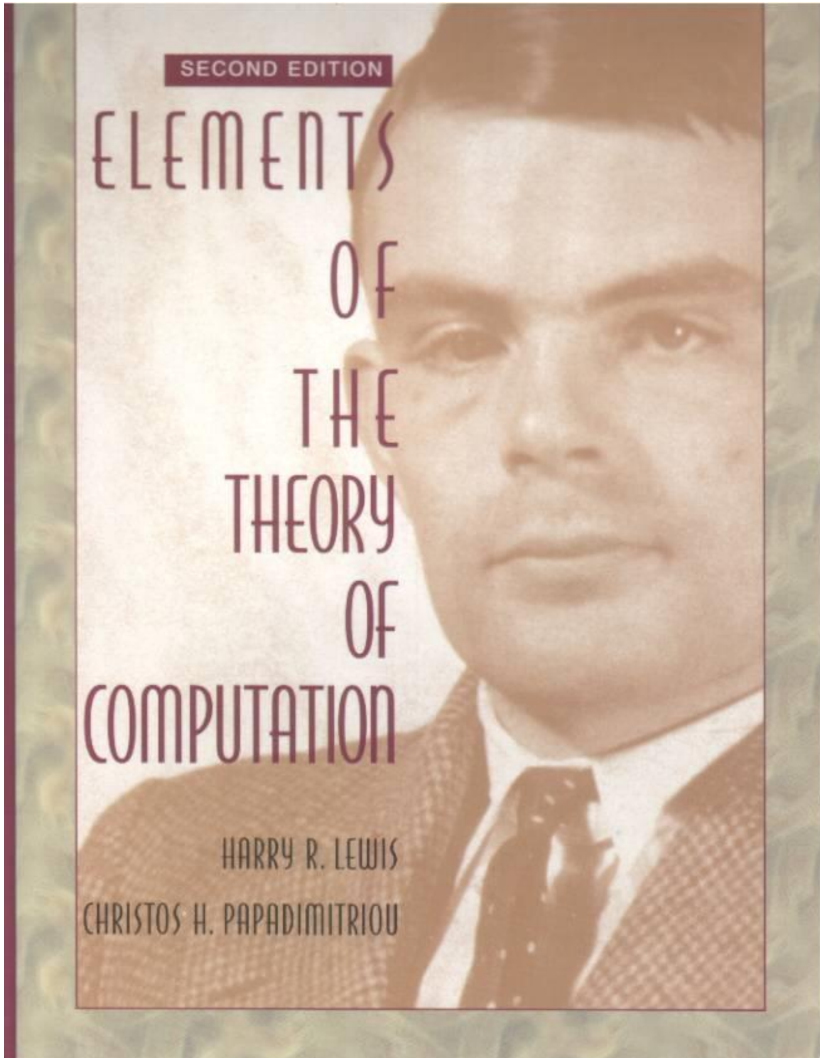
Please take advantage of my office hours and e-mail support on weeks when we do not have a tutorial. Special sessions will be scheduled before the midterm and final exam.

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

ITEM	WEIGHT
Assignments- 5	20
Midterm- Wed. June 21	30
Final exam	50

Your lowest assignment mark will be dropped before computing your assignment average.

You need an assignment average of 50% to write the final exam. Otherwise you will get N in the course.

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.

Paul Erdős

Come to class with your "brain open".

Challenge your conceptions of our definitions until you gain a complete understanding of them.

Be creative as you are solving problems.

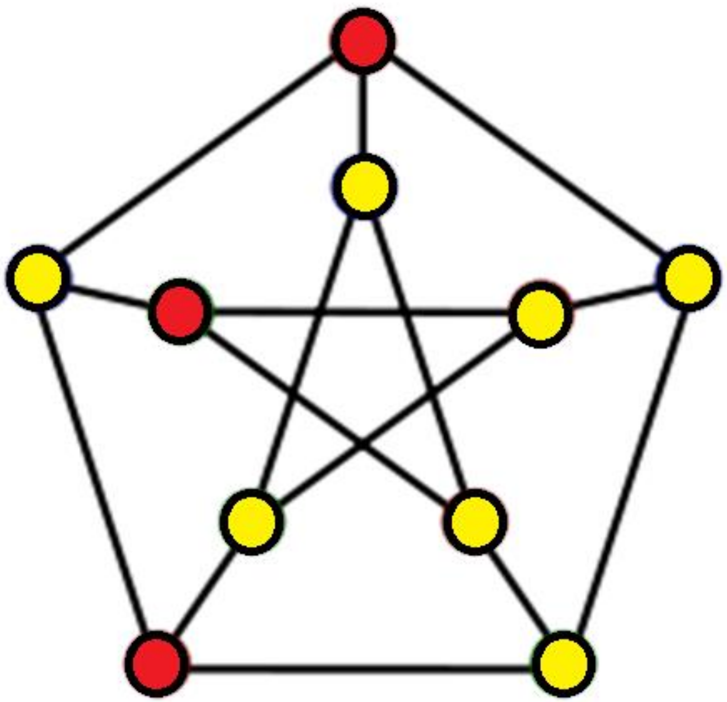
Look for answers that come from "The Book".

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.

A *dominating set* of a graph G is a subset D of the vertices of G such that every vertex v of G is either in the set D or v has at least one neighbour that is in D.
DOMINATING SET PROBLEM: Given G, k, does G have a dominating set of size k?



1. What could you use for a certificate for this problem?
2. Write pseudocode for an algorithm to check your proposed certificate.
3. How much time does your algorithm take in the worst case if you use an adjacency matrix (as a function of n, k and the maximum degree Δ)?
3. How much time with adjacency lists?

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:

{Abdulaziz, Abdulmajeed, Abdulrahman, Addie, Alex, Aria, Behnam, Bowei, Bradley, Brandon, Brendon, Cameron, Casey, Chad, Chris, Chris, Christina, Cole, Cole, Derrick, Dhaimil, Dylan, Ellie, Eric, Erik, Geoff, Graeme, Hayley, Himmat, Ian, Jake, Jason, Jeremy, JianZhao, Jiaquan, Jingjing, Jodie, Jonathan, Jordan, Jose, Justin, Kai, Kaitlin, Keifer, Kelvin, Kelvin, Kira, Kun, Leo, Liam, Liam, Lingyao, Lisa, Lok, Louis, Louis, Maston, Matt, Matthew, Maxwell, Meagan, Morgan, Nicola, Noah, Omnielle, Paul, Quintan, Rafael, Reed, Rhiannon, Rich, Richard, Rui, Sanja, Sean, Shane, Shawn, Shiyi, Siting, Sonia, Tania, Taylor, Terance, Tim, Tony, Tristan, Tyler, Tyler, Tyler, William, Yihe, Yuanfan, Yves, Zhaoxuan, Zirui}

Language: set of strings

First names of students taking CSC 320:

{Abdulaziz, Abdulmajeed, Abdulrahman, Addie, Alex, Aria, Behnam, Bowei, Bradley, Brandon, Brendon, Cameron, Casey, Chad, Chris, Christina, Cole, Derrick, Dhaimil, Dylan, Ellie, Eric, Erik, Geoff, Graeme, Hayley, Himmat, Ian, Jake, Jason, Jeremy, JianZhao, Jiaquan, Jingjing, Jodie, Jonathan, Jordan, Jose, Justin, Kai, Kaitlin, Keifer, Kelvin, Kira, Kun, Leo, Liam, Lingyao, Lisa, Lok, Louis, Maston, Matt, Matthew, Maxwell, Meagan, Morgan, Nicola, Noah, Omnielle, Paul, Quintan, Rafael, Reed, Rhiannon, Rich, Richard, Rui, Sanja, Sean, Shane, Shawn, Shiyi, Siting, Sonia, Tania, Taylor, Terance, Tim, Tony, Tristan, Tyler, William, Yihe, Yuanfan, Yves, Zhaoxuan, Zirui}

This set is the same as the previous one.

Language: set of strings

Strings over {a,b} with even length:
{ε , aa, ab, ba, bb, aaaa, aaab, aaba, …}

{ Strings over {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} that represent prime numbers}

= {2, 3, 5, 7, 11, 13, 17, 19, 23, …}

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?

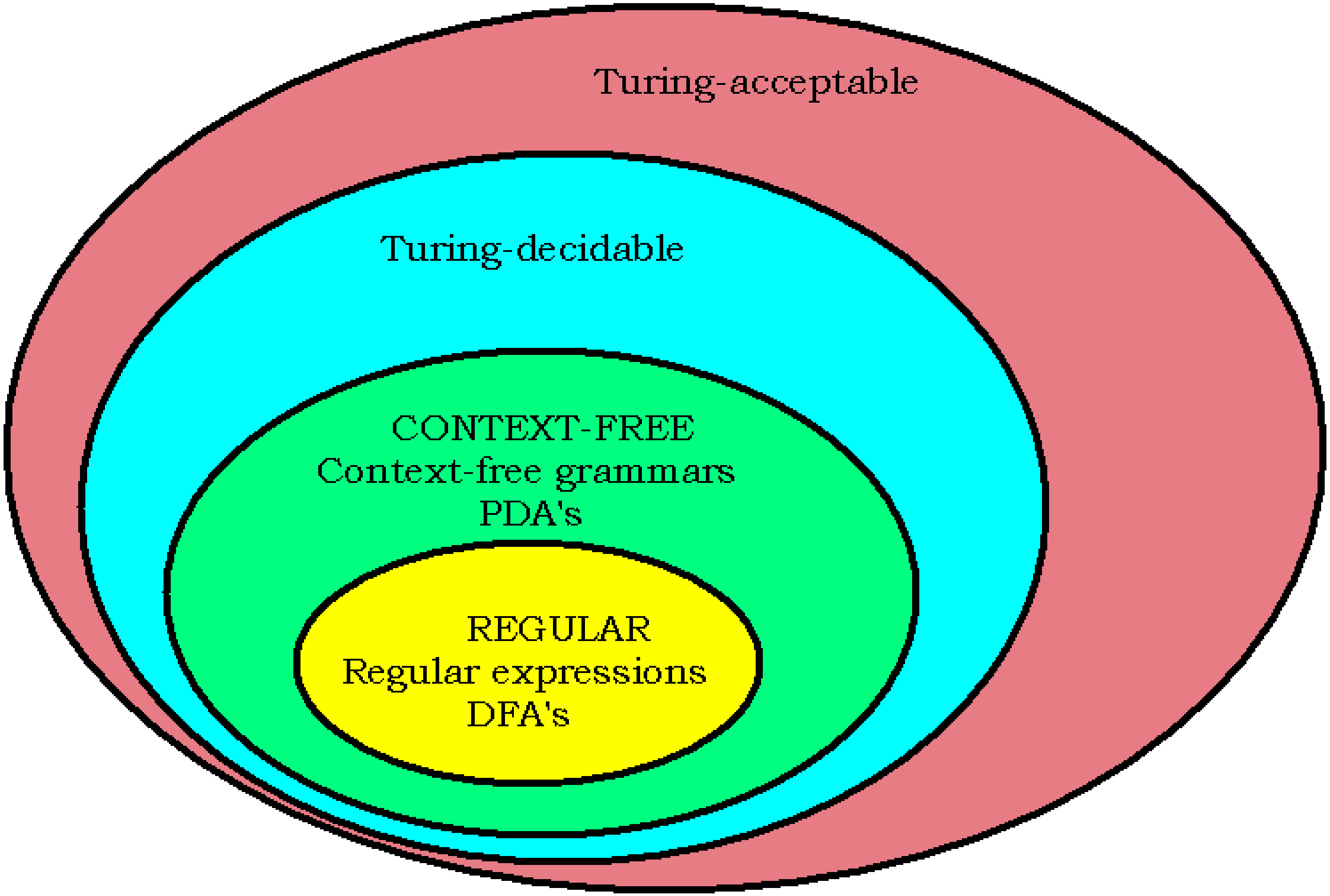
Regular languages: finding patterns in strings

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

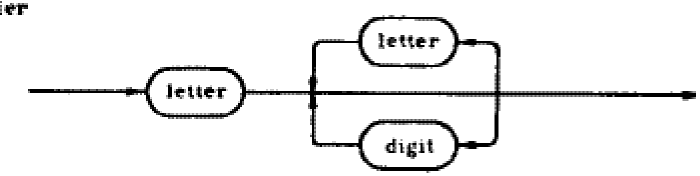
Turing-decidable languages: yes/no questions which are computable on a computer

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

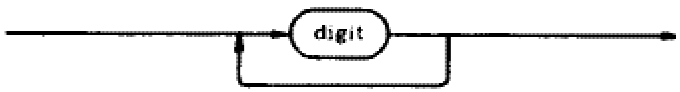
Classes of Languages



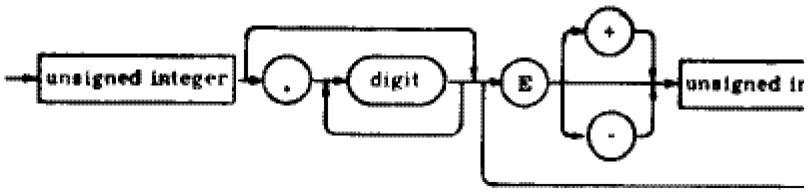
identifier



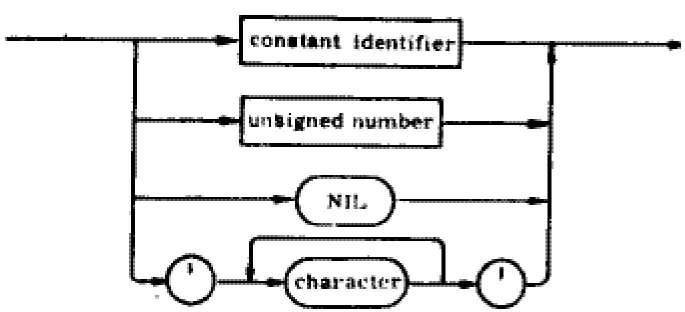
unsigned integer



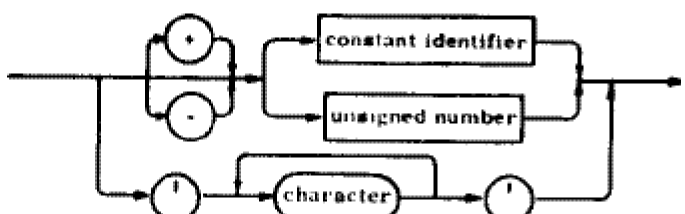
unsigned number



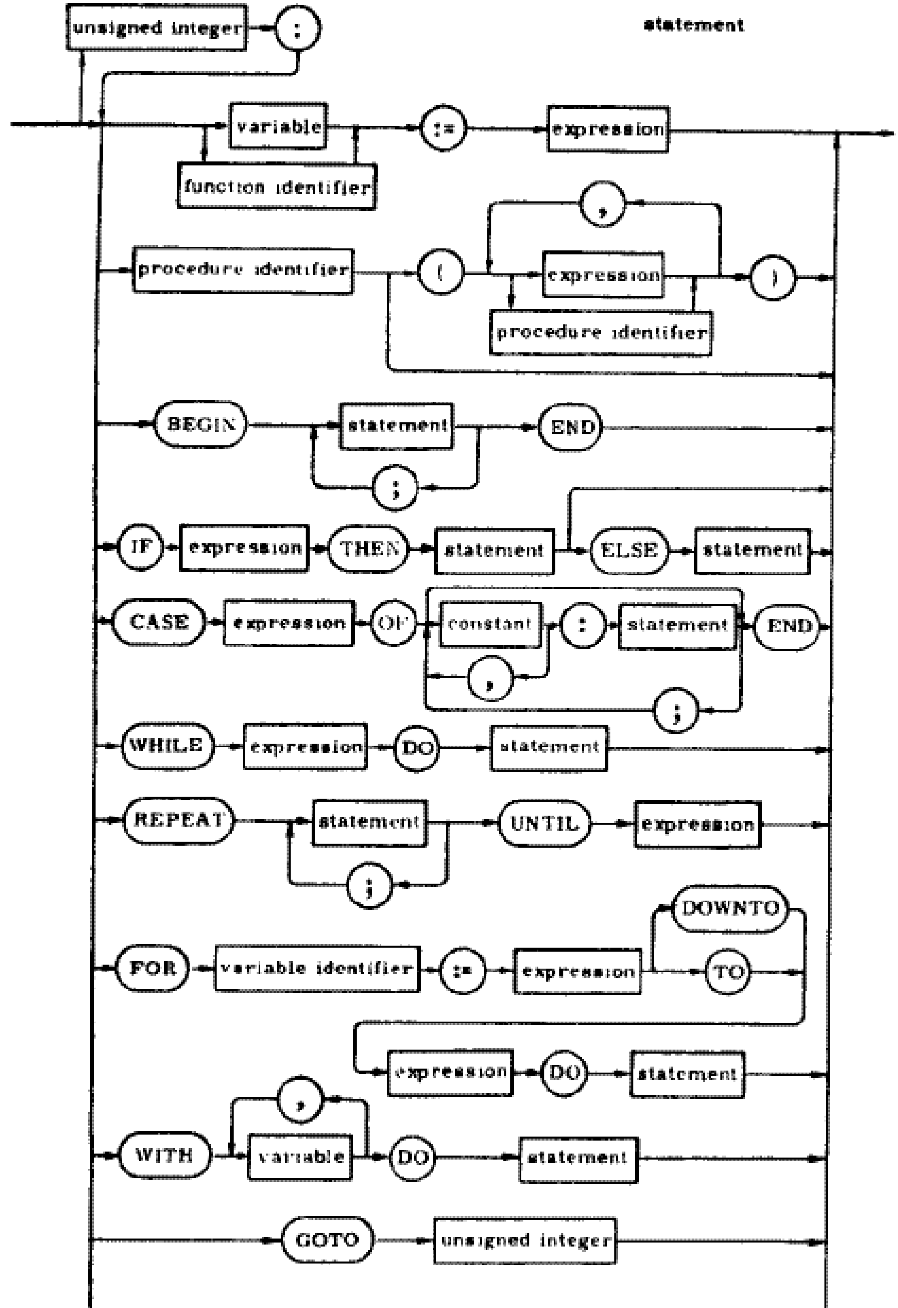
unsigned constant



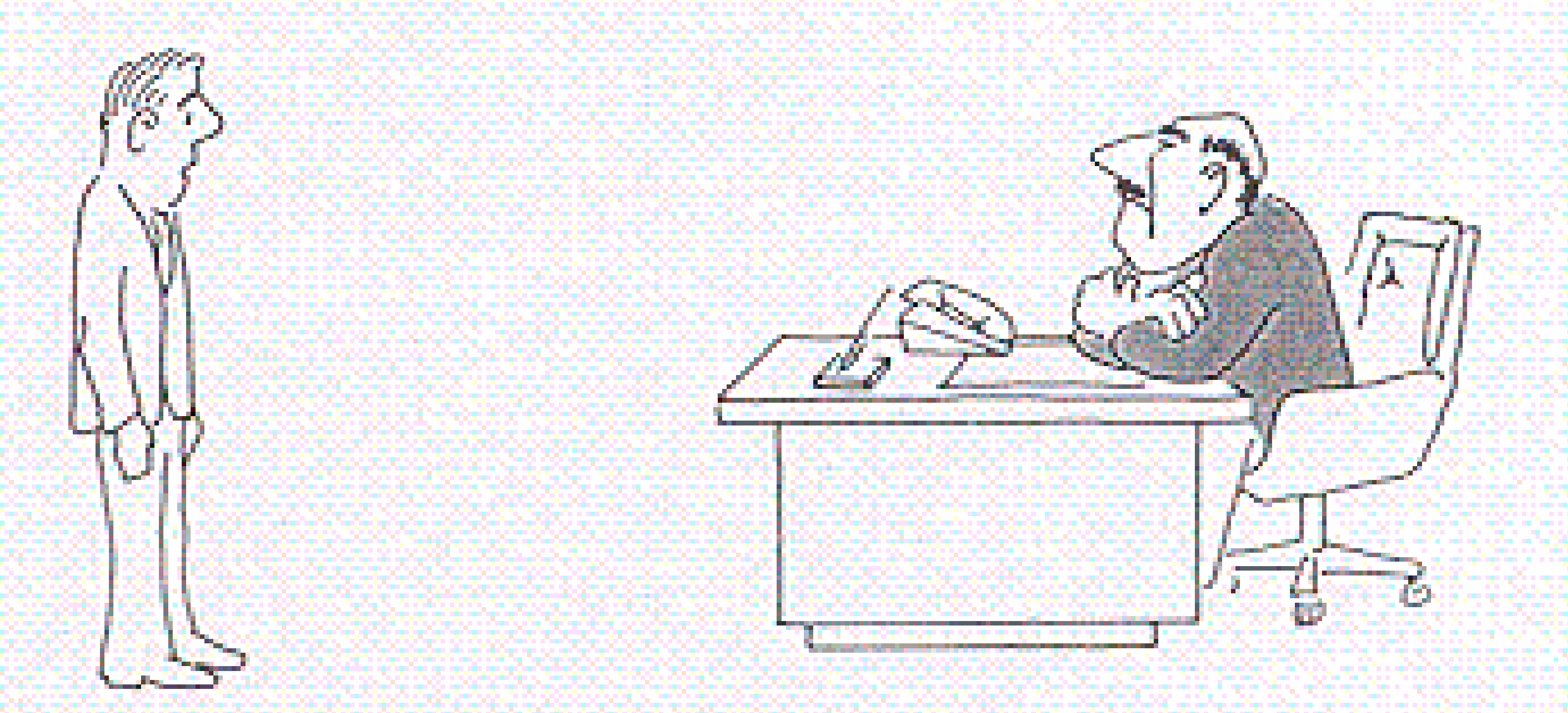
constant



statement



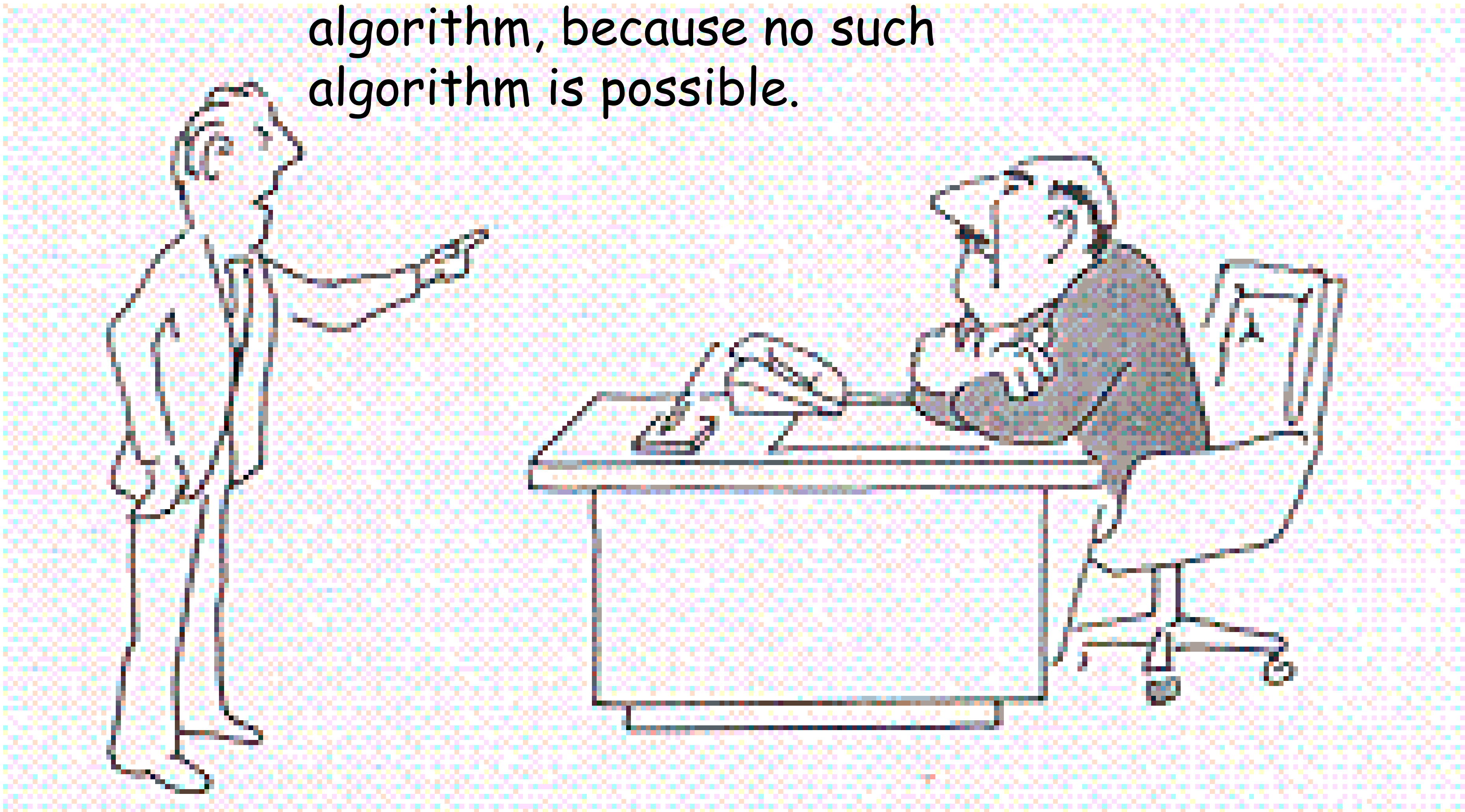
NP-completeness

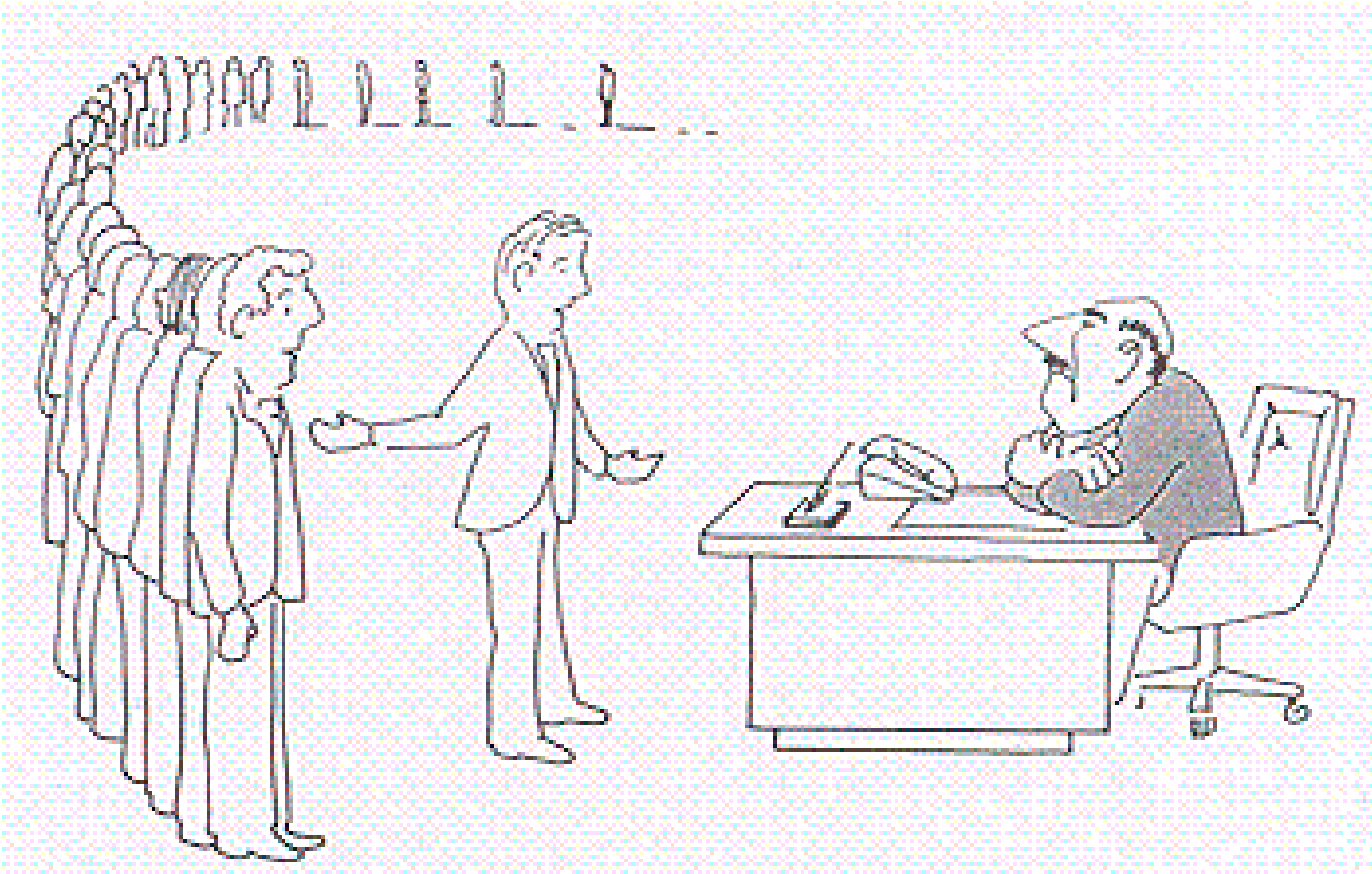


I can't find an efficient algorithm,

I guess I'm just too dumb.

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

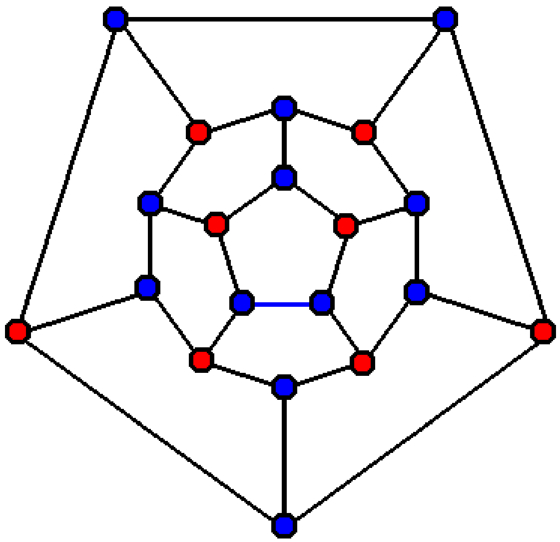




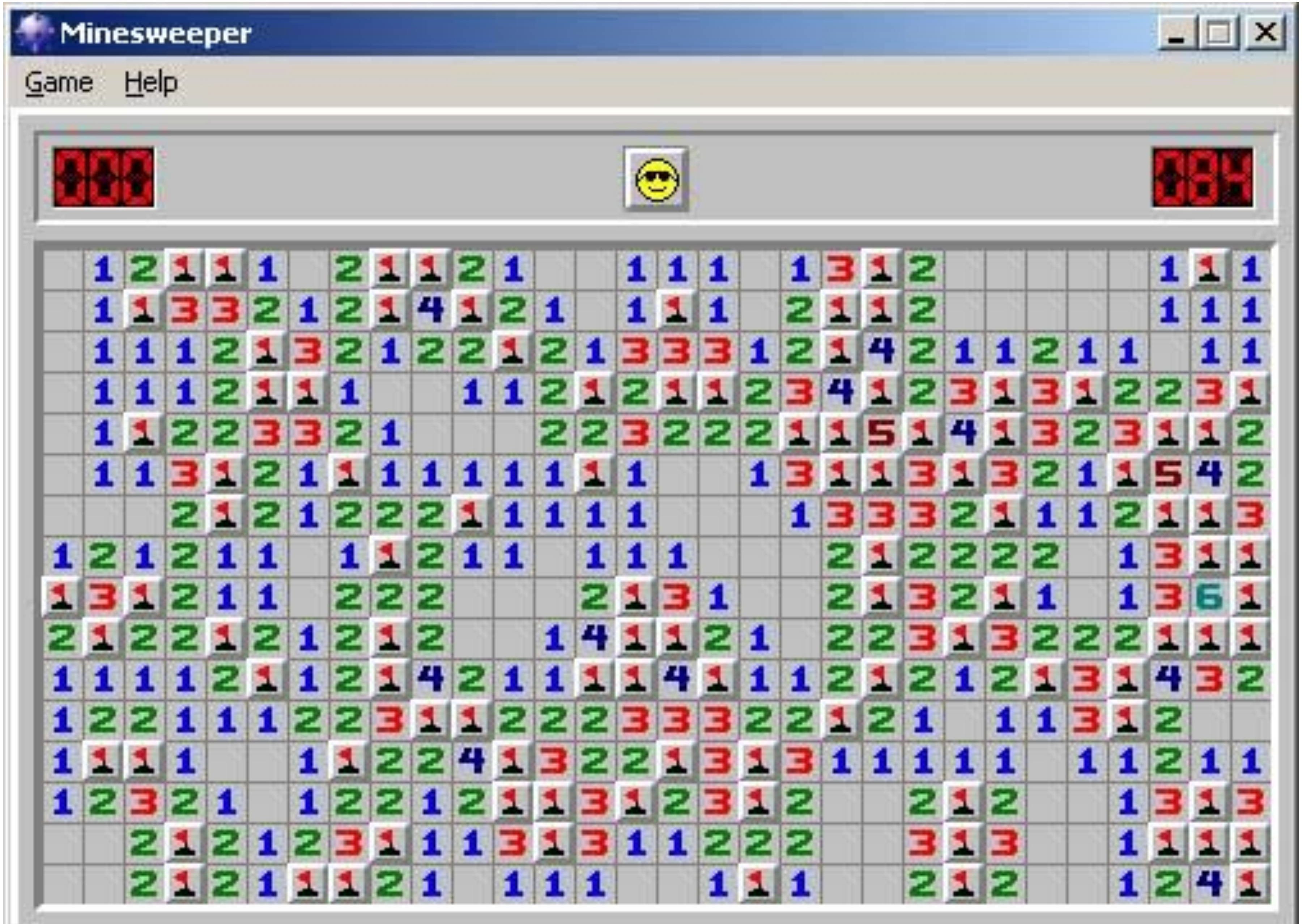
I can't find an efficient algorithm, but neither can all these famous people.

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.