Proof of the Day:

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

For example:

The 8 binary strings of length 3 are:

000, 001,

010, 011,

100, 101,

110, 111

Note that $8 = 2^3$.

Does your base case mention strings?

The base case could either be:

The number of binary strings of length 0 is one because the empty string "" is the only binary string of length 0. The formula gives $2^0 = 1$ as required.

When you take CSC 320 you will need to know about the empty string.

The base case I would expect CSC 225 students to give is:

The number of binary strings of length 1 is two because 0 and 1 are binary strings of length 1. The formula gives $2^1 = 2$ as required.

Your induction step should start with:

Assume that the number of binary strings of length k is 2^k.

We want to prove that the number of binary strings of length k+1 is 2^{k+1}.

In your proof:

Give yourself a mark of 0 if you just scribbled some algebra and did not talk about strings at all.

Take off some marks if you did not clearly indicate where you applied the induction hypothesis.

The 8 binary strings of length 3 are:

000, 001,

010, 011,

100, 101,

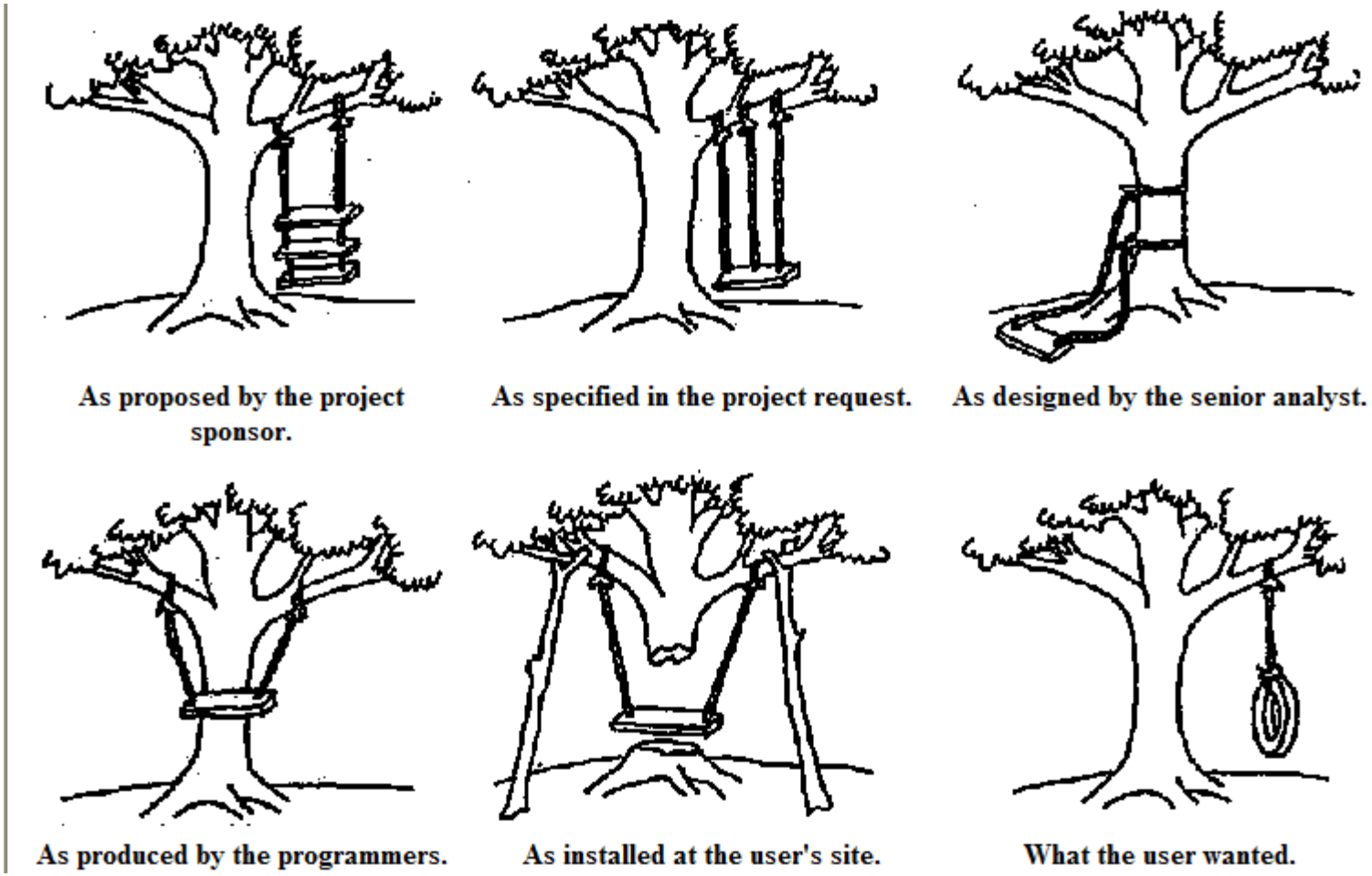
110, 111

Inductive definition of a binary string of length k:
[Basis] The empty string is a binary string of length 0.

[Induction step] If w is a binary string of length k then w 0 and w 1 are binary strings of length k+1.

CSC 225: Design and Analysis of Algorithms

Dr. Wendy Myrvold, ECS 552, wendym@uvic.ca



Cartoon from: http://www.uoregon.edu/~ftepfer/SchlFacilities/TireSwingTable.html

Announcements

Assignment #1:

Part 1A Programming Questions: Upload to connex by before Saturday Sept. 23 at 11:55pm

Part 1B Written Questions: Hand in on paper at the beginning of class on Thursday September 28. To be posted soon.

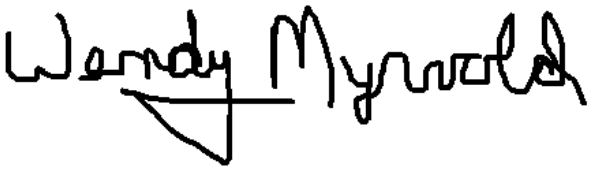
It's better to hand in a partially completed assignment then to hand in nothing at all. You must pass the assignments to pass the course.

Make sure you sign the attendance sheet today to get credit for attending.

Powerpoint slides will be posted: click on the "Selected class notes" link on the course web page.

Students are expected to attend all the classes. In order to write the final exam, students should not miss more than 6 classes with the exception of extreme circumstances with appropriate documentation (for example, a note from a health care provider).

Your signature on the attendance sheet must be legible:



OK

NOT OK

If you want credit for attendance I should be able to read your name from what you have written.

DO NOT SIGN the sheet for a friend.

Lab #1 is posted: Your first lab will be Sept. 11, 12, or 15.

Try to complete the exercises posted before attending the lab. Attend the lab even if you do not find time to complete everything beforehand.

Lab sections:

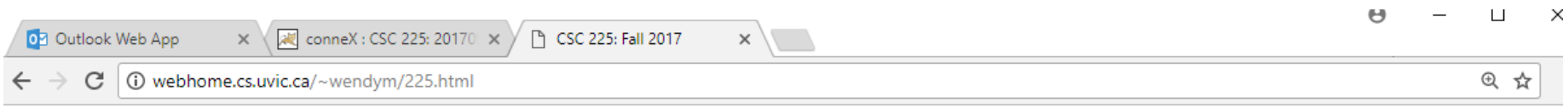
B01	ECS 258	M	13:30-14:20
B02	ECS 258	M	14:30-15:20
B03	ECS 258	M	15:30-16:20
B04	ECS 258	T	09:30-10:20
B05	ECS 258	T	10:30-11:20
B06	ECS 258	F	13:30-14:20
B07	ECS 258	F	14:30-15:20

Class Materials

Connex: calendar, electronic assignment submission (for programs), feedback on electronic submissions, 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.

http://webhome.cs.uvic.ca/~wendym/225.html



Home page for CSC 225
Algorithms and Data Structures
Fall 2017

This page provides the links to useful resources for CSC 225 students.

Office hours:

MWR 3:30-4:15pm, also I can stay after class if there are questions.

Help is also available by sending me e-mail (wendym@cs.UVic.ca), or by appointment. If you would like an appointment, send e-mail (wendym@cs.UVic.ca) with the times you are available, and I will choose a mutually convenient meeting time. Note that I answer ALL e-mails from students taking my classes. If you do not get a response within a reasonable time, please let me know. Include 225: and an informative subject in your messages to me.

Course information

1. Official Course Outline.
2. Unofficial course outline.
3. Selected class notes.
4. Some old midterms
5. Some old final exams
6. Final exam study aid.
7. Foundations of Computer Science by Aho and Ullman
   has excellent chapters on:
   Iteration, Induction, and Recursion
   The Running Time of Programs
   Click here if the link above is not working.

http://webhome.cs.uvic.ca/~wendym/225.html

The labs/assignments reside on the class web page:

Labs

1. Lab #1: Sept. 11, 12 or 15.

Assignments

If you are sick or have family commitments when an assignment is due, then some options for assignment submission are:

1. Slip it under my office door before the deadline.
2. Ask a friend to hand it in. It is recommended that you place it in a large envelope which is sealed with your signature across the seal to avoid potential copying.
3. Send e-mail to me: wendym@uvic.ca.

Note: on some browsers you may have to hit reload or refresh to get the latest version of a web page.

1. Assignment #1A: Programming questions: Upload your solution to connex by 11:55pm on Saturday Sept. 23.
2. Assignment #1B: Written questions: coming soon.

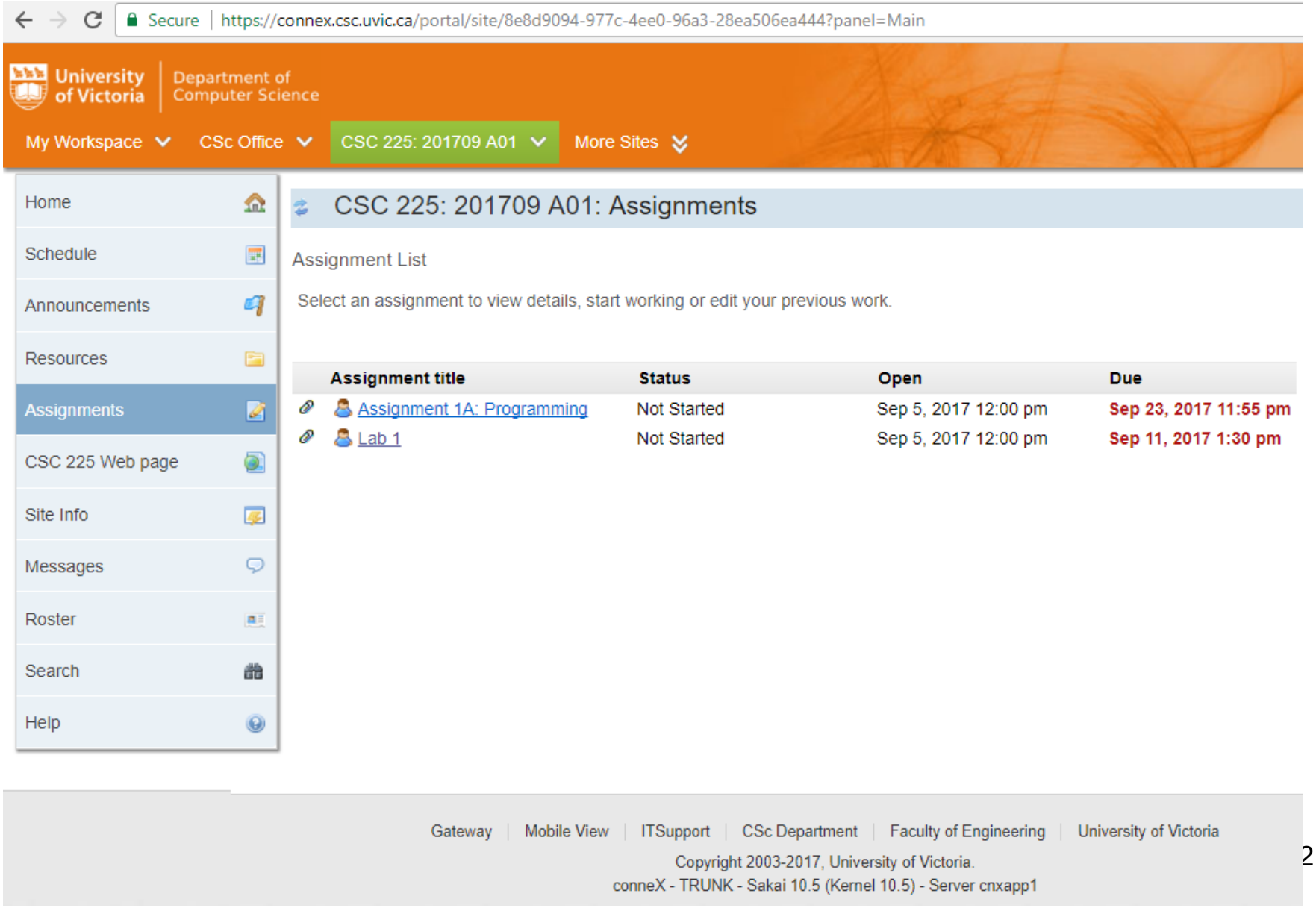
Answers to some student questions

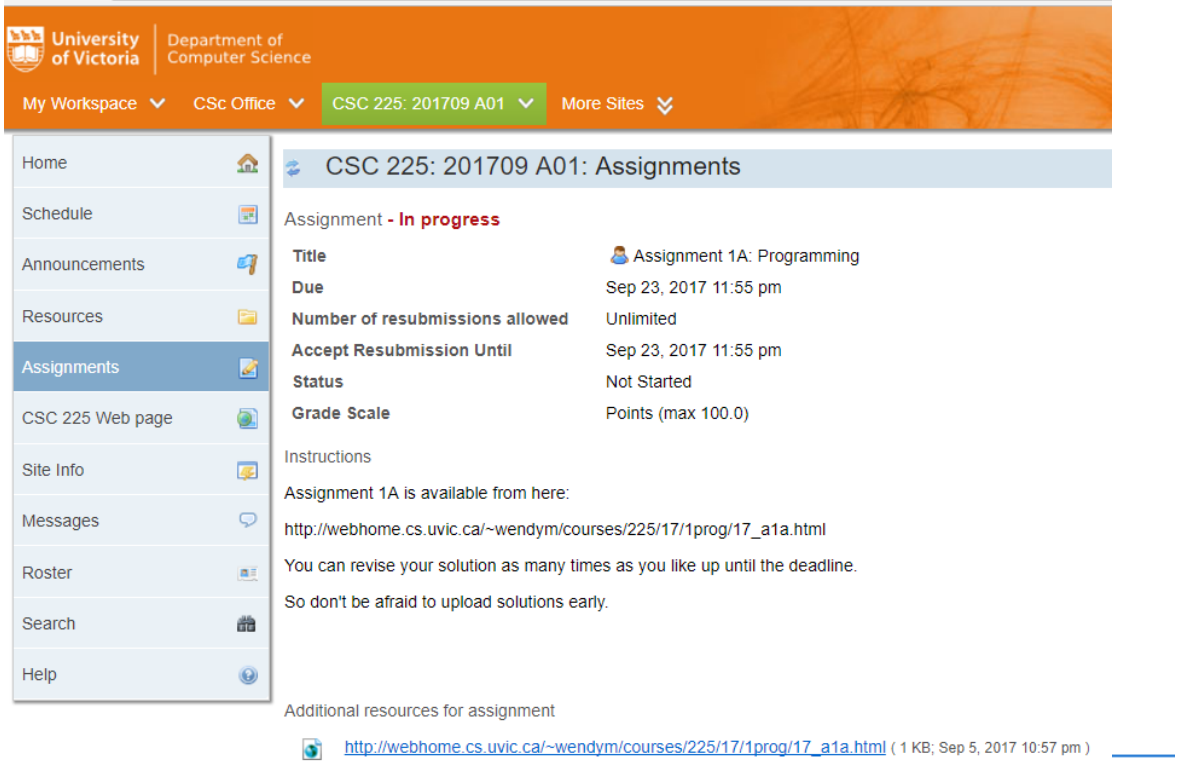
So far, I have not had any questions.

Return to TOP of the page.

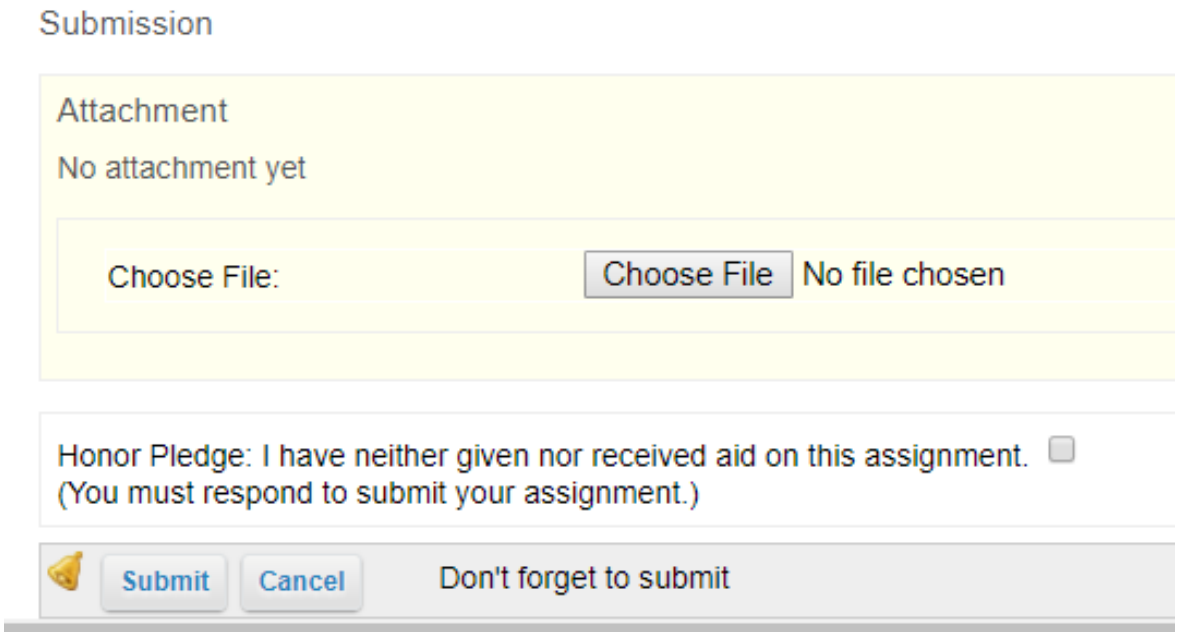
CSC 225 Home Page / maintained by Wendy Myrvold / wendym@cs.UVic.ca / revised Sept. 6, 2017

Lab 1 and Assignment 1A are also available on connex:





You can upload the file as many times as you like until the deadline.



You can upload your programs at the bottom of the assignment page. Submit 1 file:

LinkedList.java

Outline for Lecture 1

- Who is the instructor?
- My research interests
- Logistics for CSC 225- the critical points are included on the course outline
- Brief overview of course content- don't worry about taking notes today

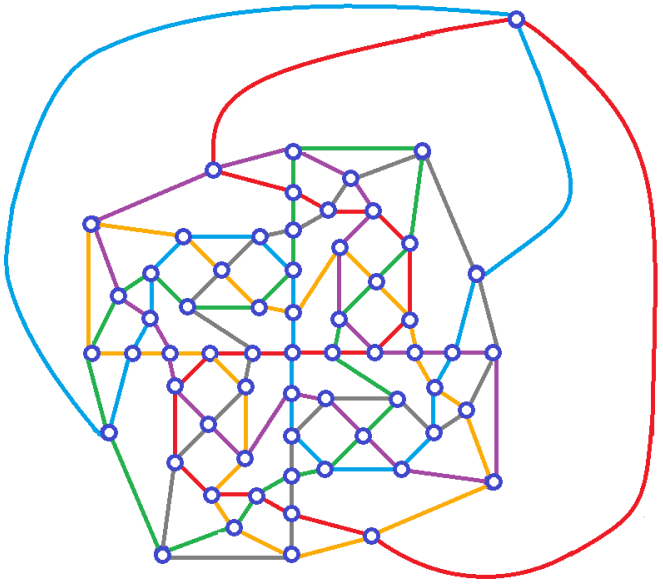
About me:

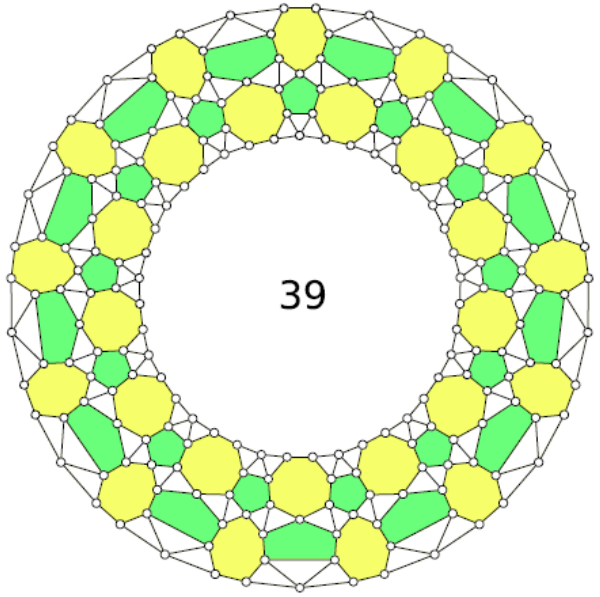
B.Sc. : Computer Science, McGill University, 1983

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





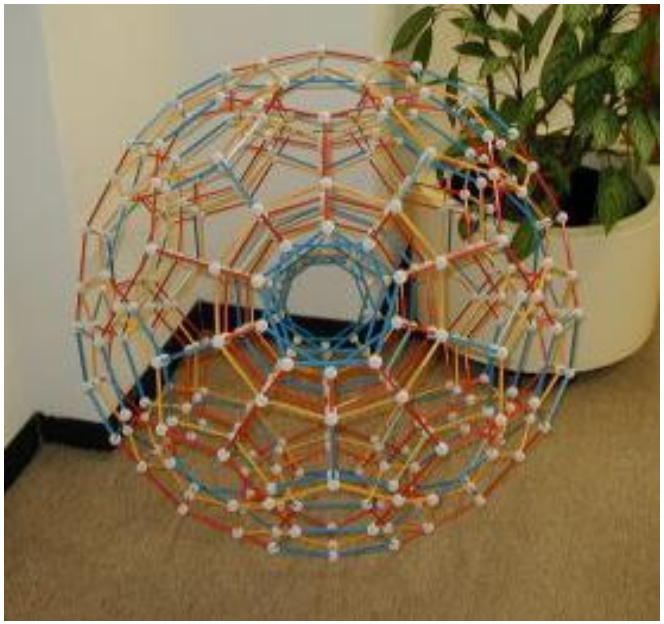
Bring your parents to work day at Google.

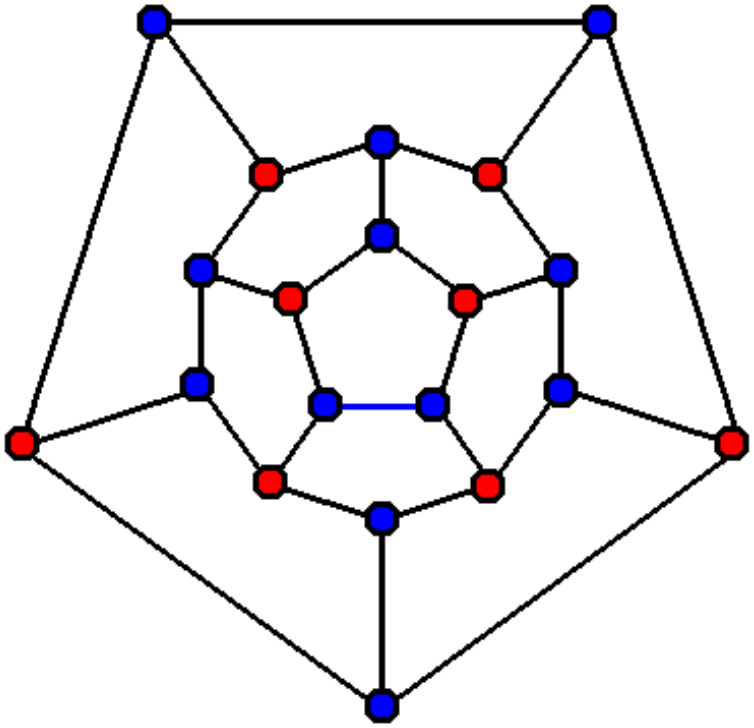

M
A
CCM

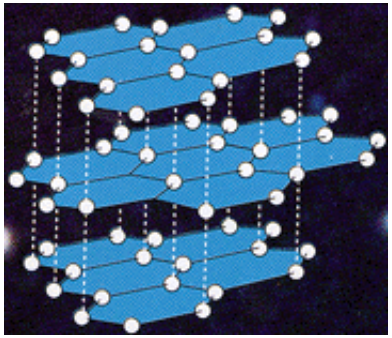
My Research: Large Combinatorial Searches

Independent Set:

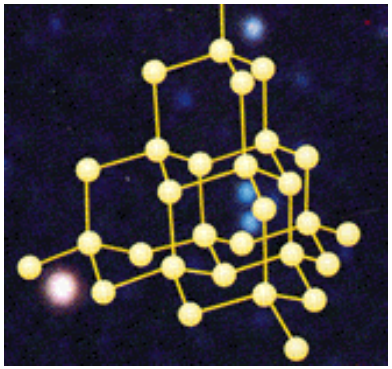
Set of vertices which are pairwise non-adjacent



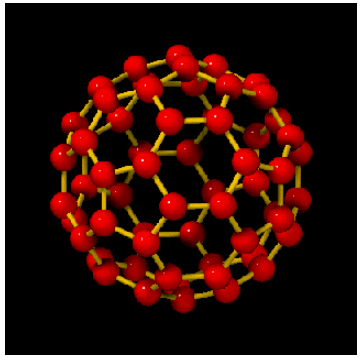




Graphite



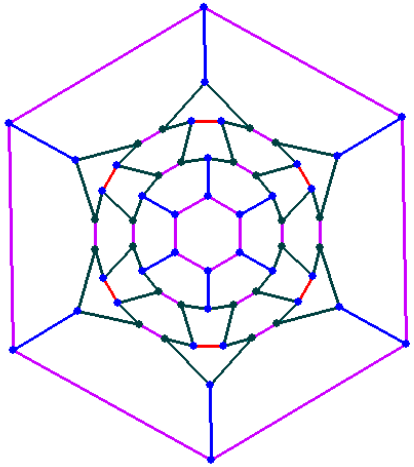
Diamond

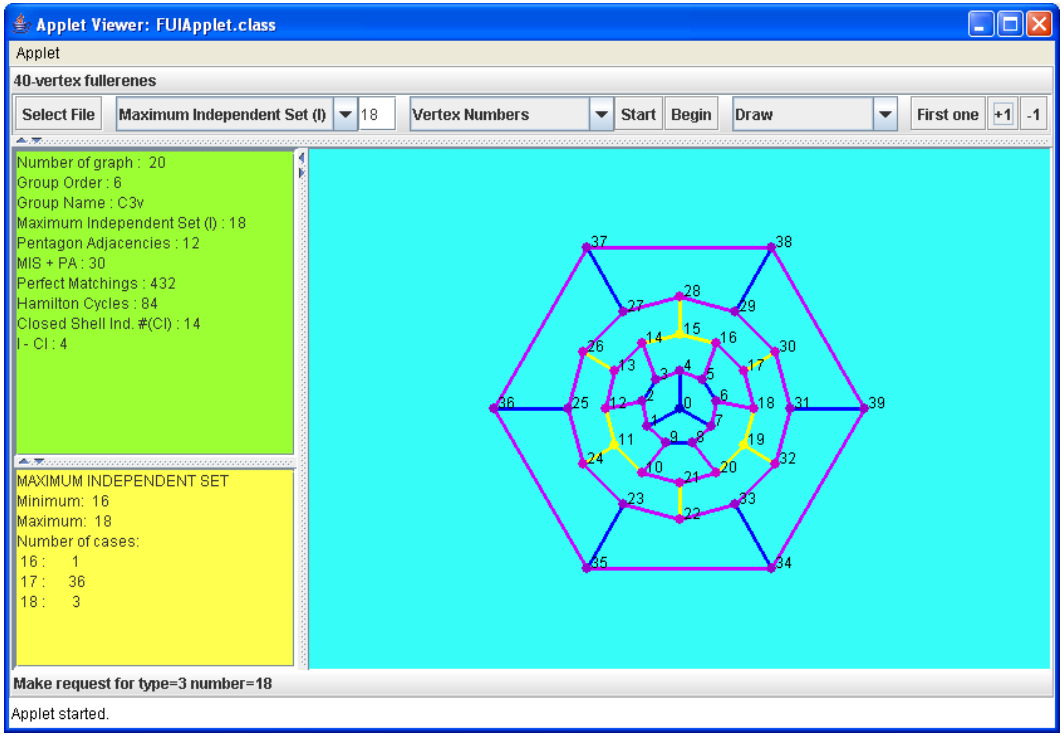


Fullerene

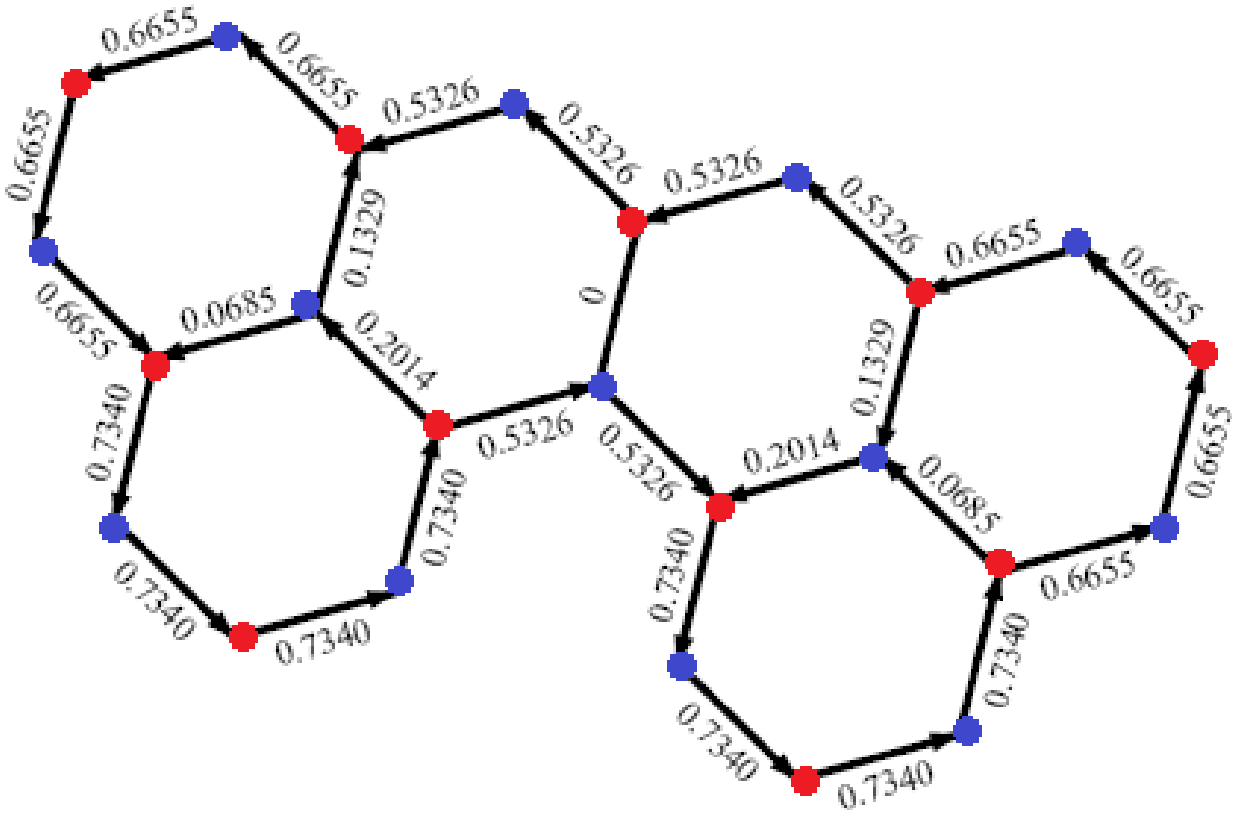
Fullerenes:

Working with Patrick Fowler (chemist)



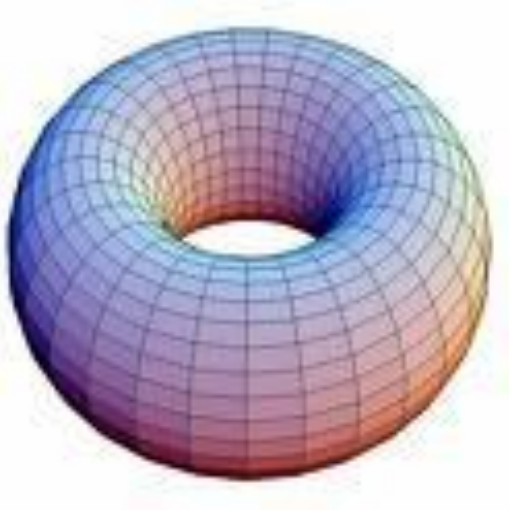


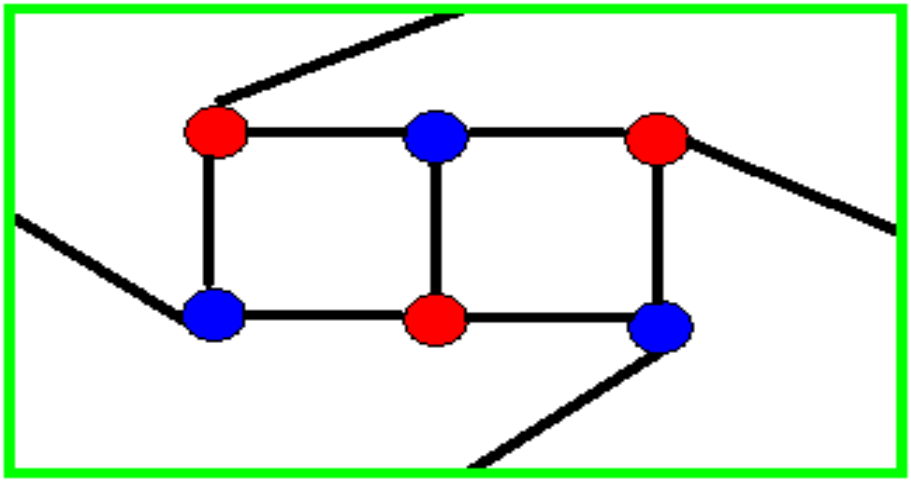
Currents in benzenoids (joint work with Patrick Fowler, a chemist):

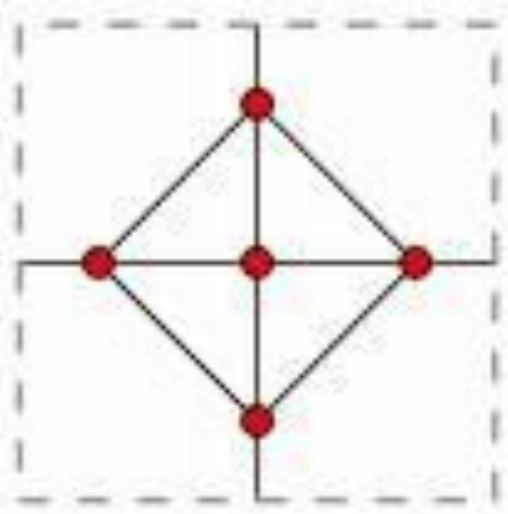


Topological Graph Theory:

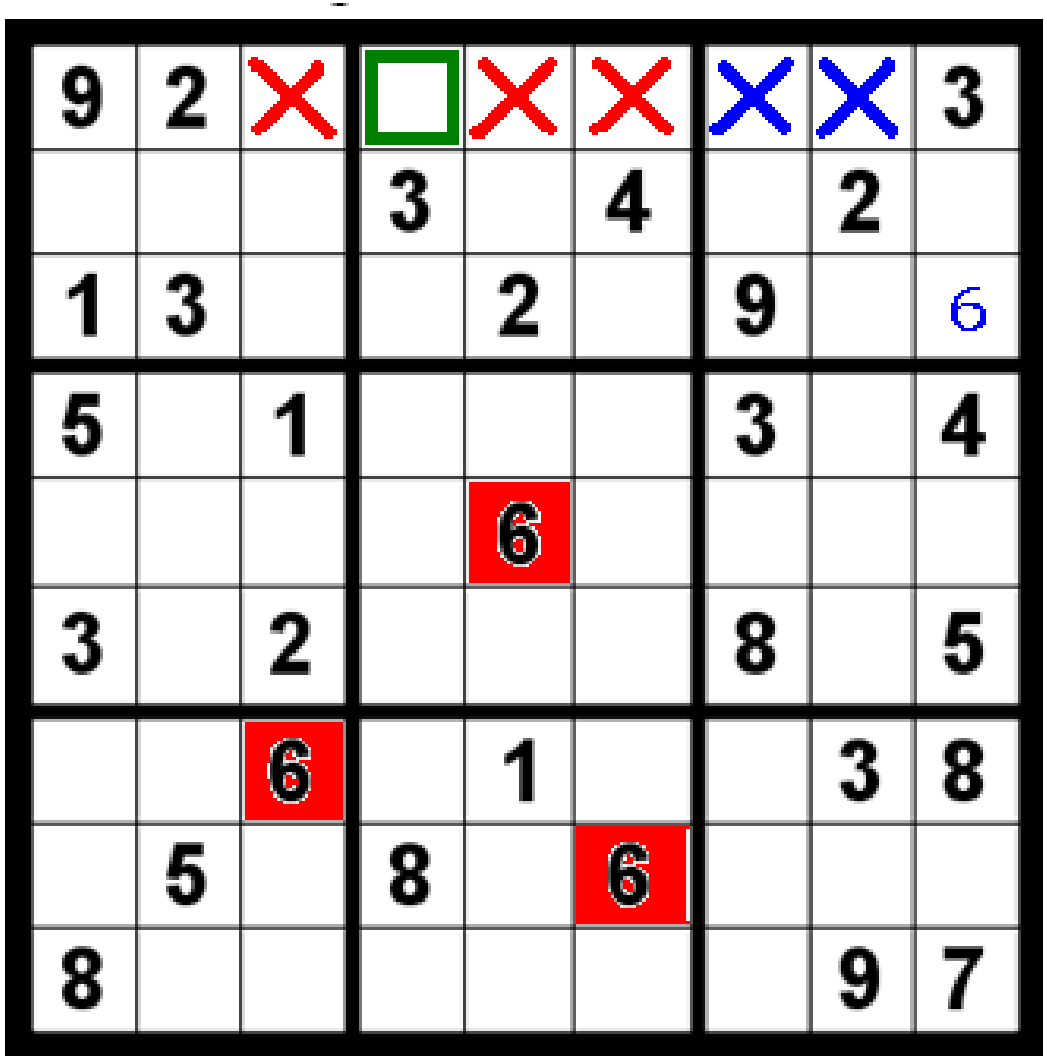
Algorithms and Obstructions





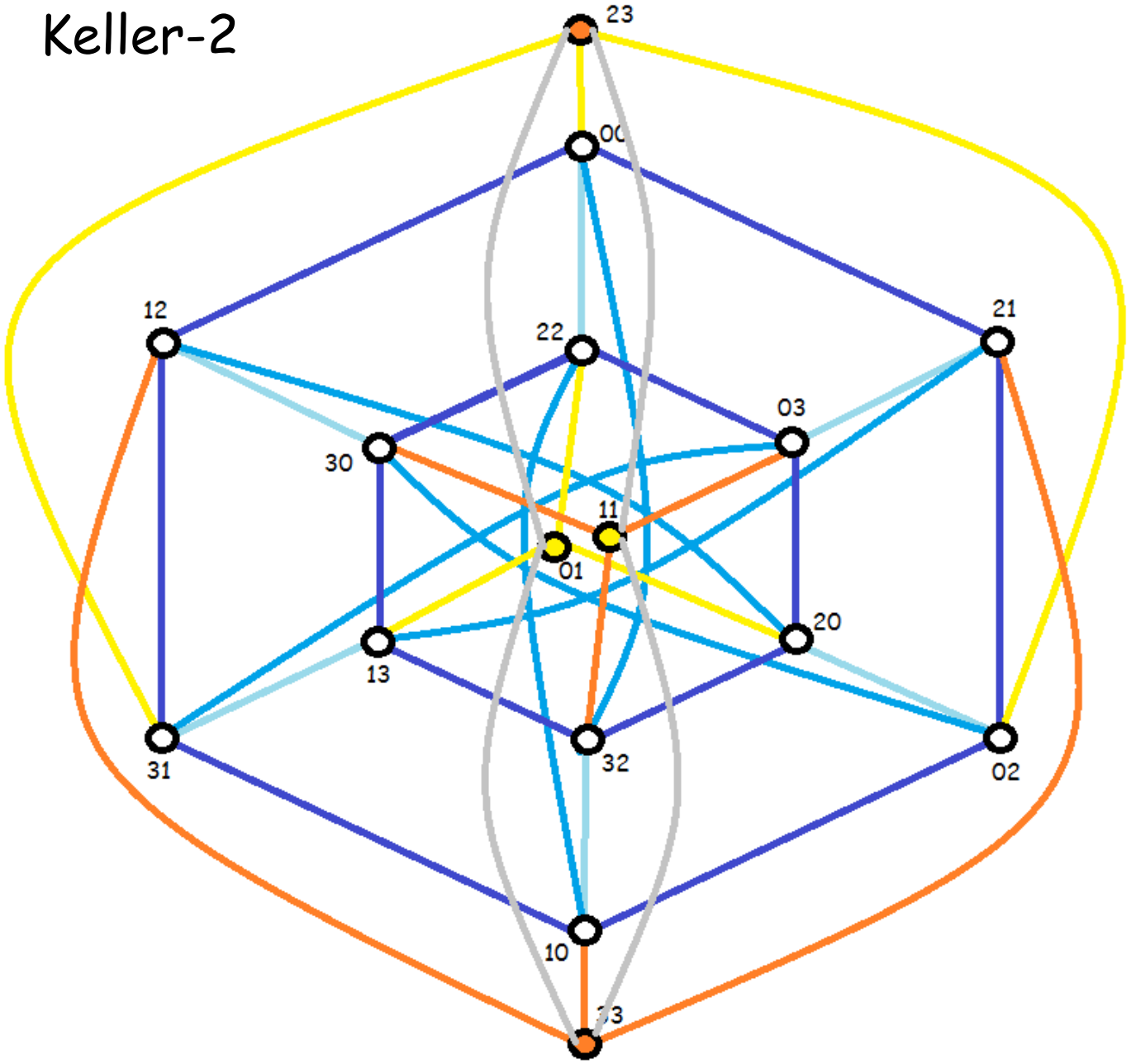


Latin Squares



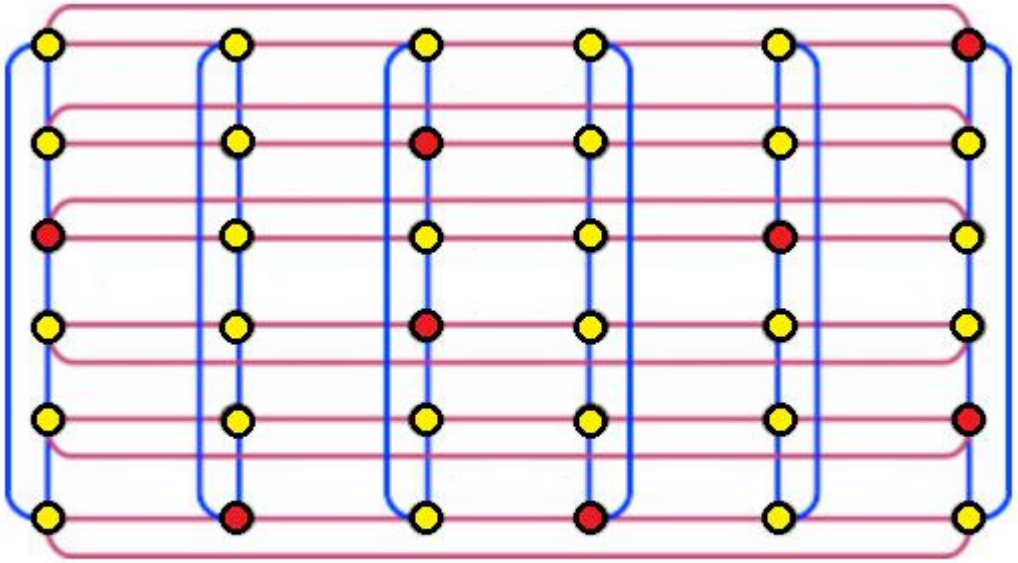
Please come talk to me if you are looking for Honours project research topics or for an NSERC undergraduate research project.

Keller-2

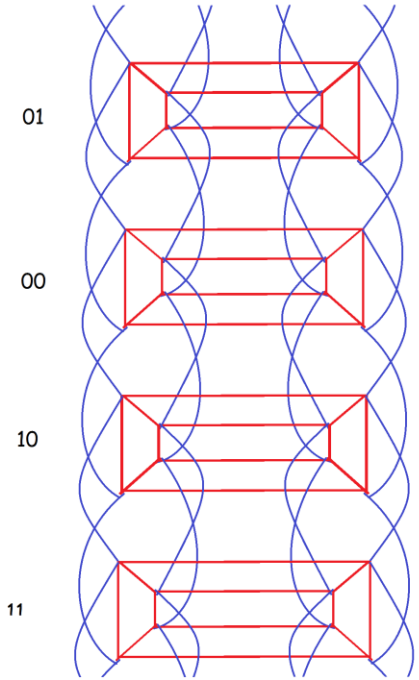


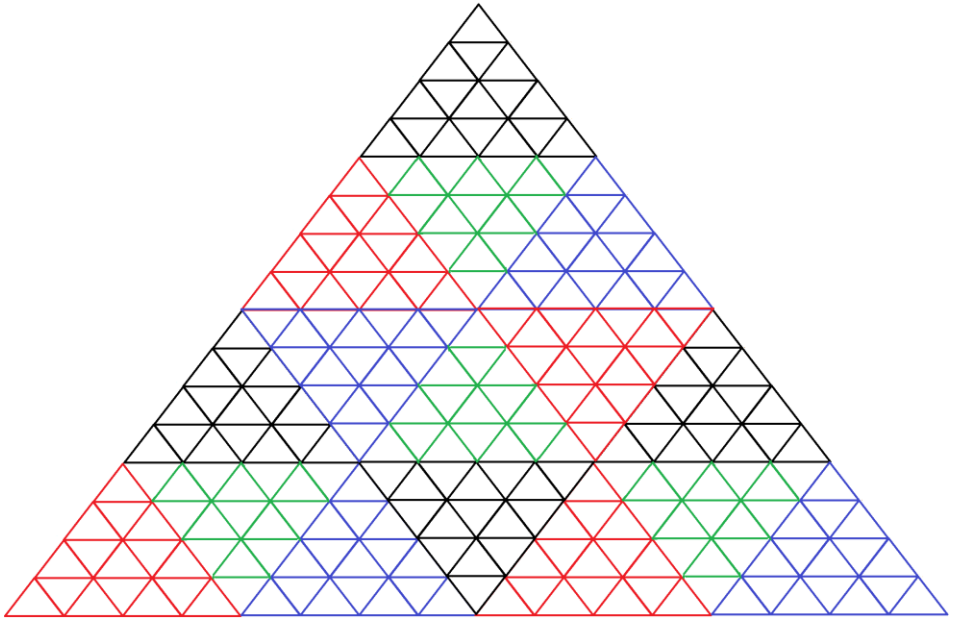
Found the maximum clique order in Keller-7.

Interested in coloring the complements of Keller-5, Keller-6, and Keller-7.



A subset D of the vertices of a graph is a dominating set if every vertex of the graph is either in D or has a neighbour that is in D.





Open problems: what is the size of a minimum dominating set for a hypercube of dimension 10 (1024 vertices)? For larger triangular grid graphs?

CSC 225 Logistics

Course Website: http://webhome.cs.uvic.ca/~wendym/225.html

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

Office: ECS 552
Phone Number: 472-5783 (use e-mail for a faster response)
Office Hours: MWR 3:30pm or after class. I will stick around after class until nobody has an more questions.

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.

Lecture Schedule: MWR 4:30-5:20pm, ECS 123

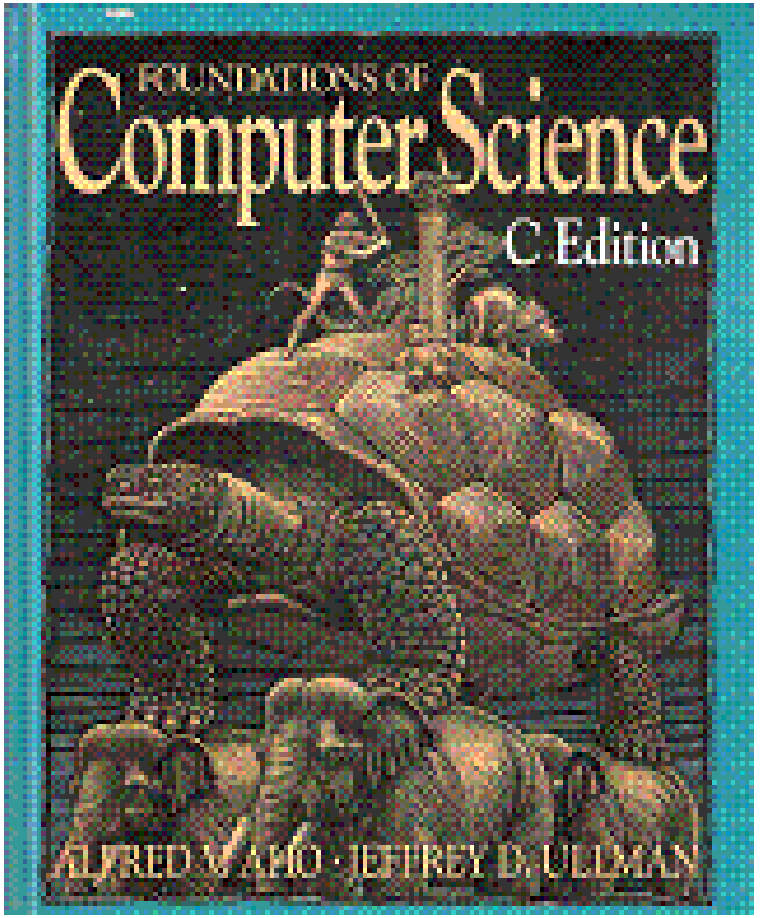
Projected Tutorial Schedule (total 7):
[Available from course web page]

Lab #1: Monday Sept. 9.
Lab #2: Monday Sept. 16.
Lab #3: Monday Sept. 23.
Lab #4: Monday Sept. 30.
Lab #5: Monday Oct. 21.
Lab #6: Monday Nov. 18.
Lab #7: Monday Dec. 2.

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.

"This book has been taken out of print by W. H. Freeman. You are welcome to use it if you like. We believed in 1992 it was the way to introduce theory in Computer Science, and we believe that today."

- Al Aho and Jeff Ullman



This book has excellent material on induction and Big Oh notation.

http://infolab.stanford.edu/~ullman/focs.html

Course Textbook:

Great pictures.
Elegant Java examples.
Theoretical precision.
Covers class material.
Also used for CSC 226.

Algorithms, Fourth edition,
by Robert Sedgewick and Kevin Wayne,
Addison-Wesley, Toronto, 2011.

CSC 225 Grading

Written assignments	5	15%
Programming assignments	3	15%
Midterm	Thursday Oct. 19, in class	20%
Final Exam	Scheduled by registrar	50%

You need a passing assignment average to write the final exam. Otherwise, your grade is N.

Keys to Success

Attend all classes and labs.

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" and phone/facebook/e-mail/video games off.

Try to understand what we are doing instead of memorizing things. Take notes.

Be creative and make sense of things as you are solving problems.

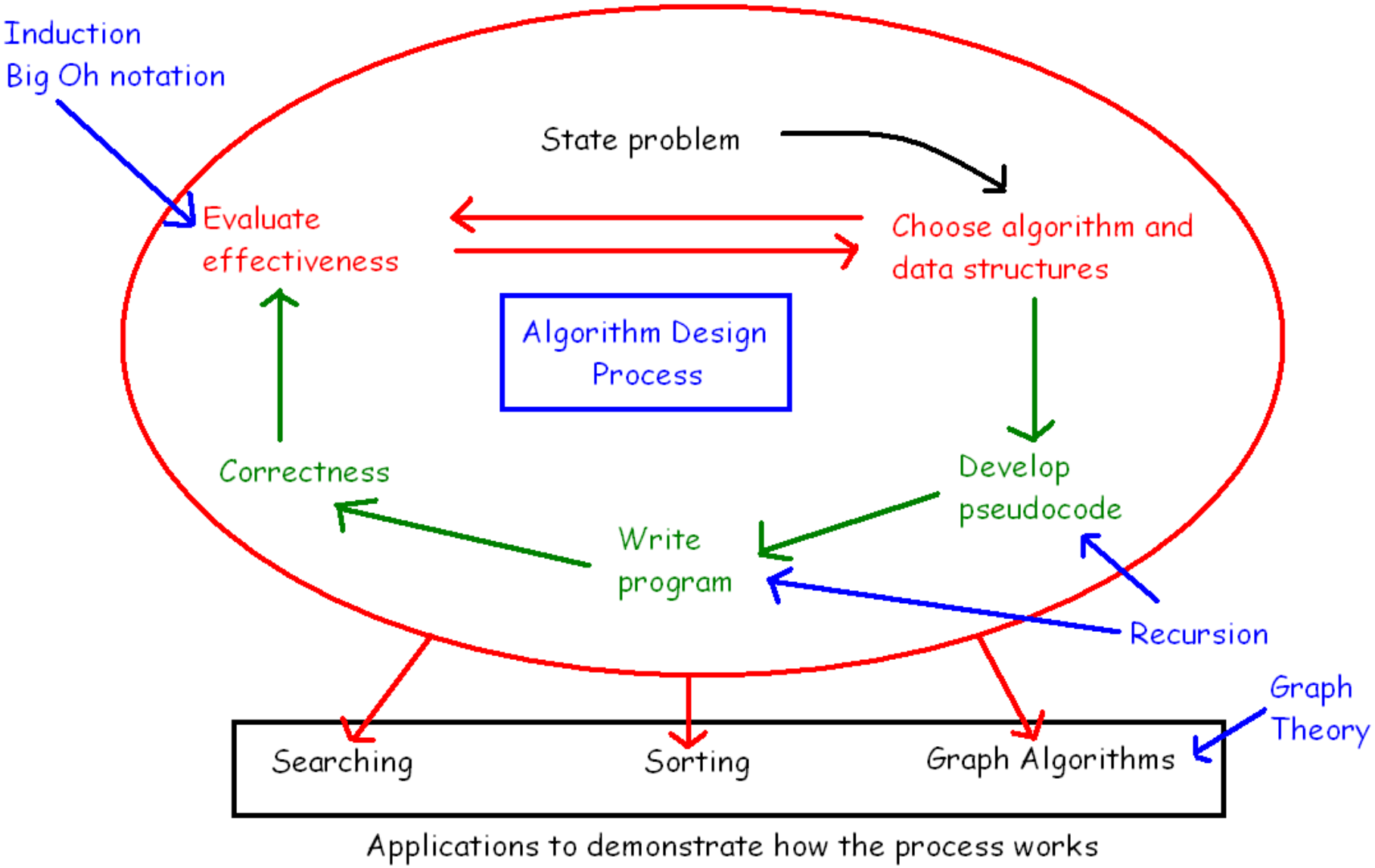
Look for answers that come from "The Book".

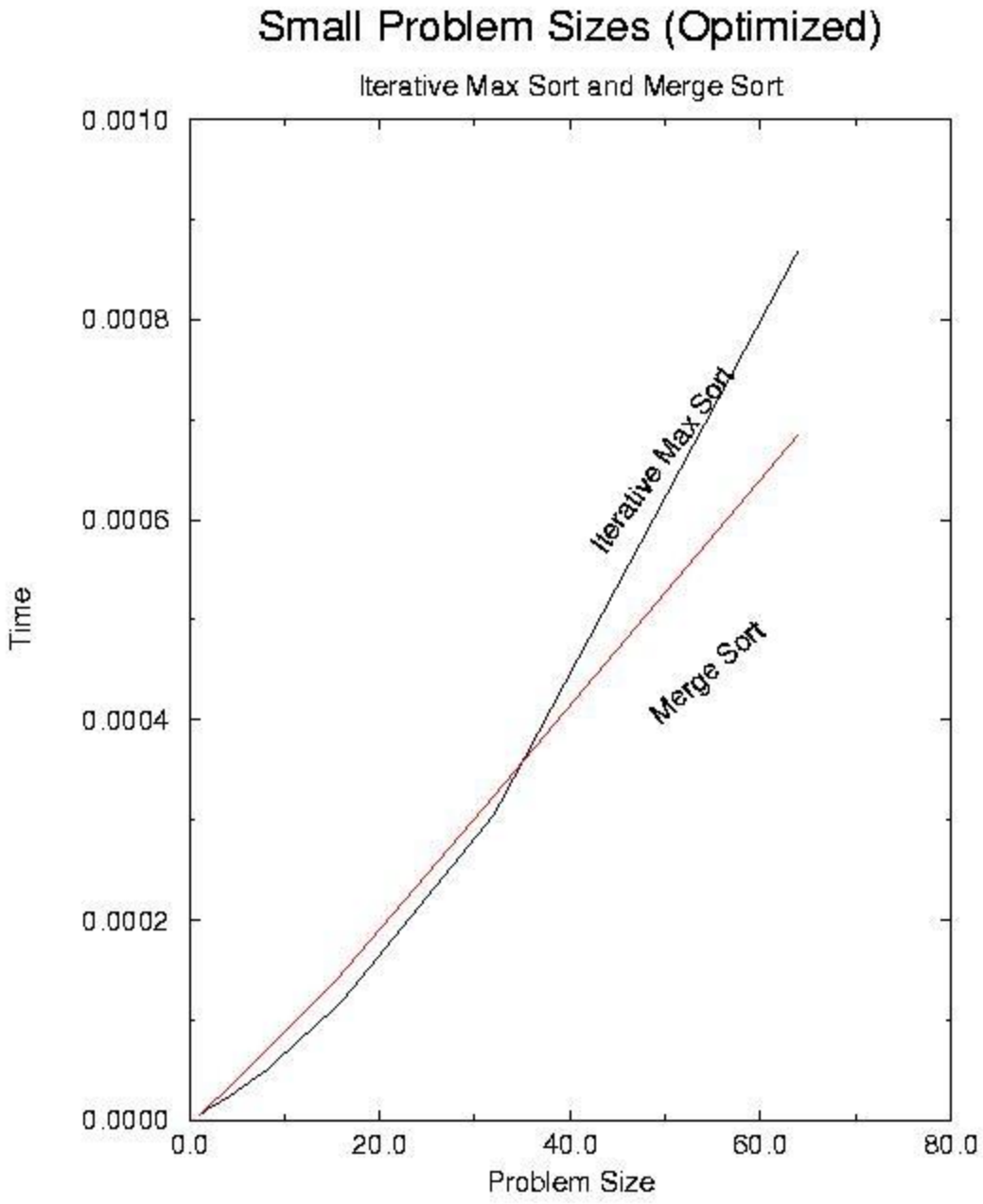
Students with a disability

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

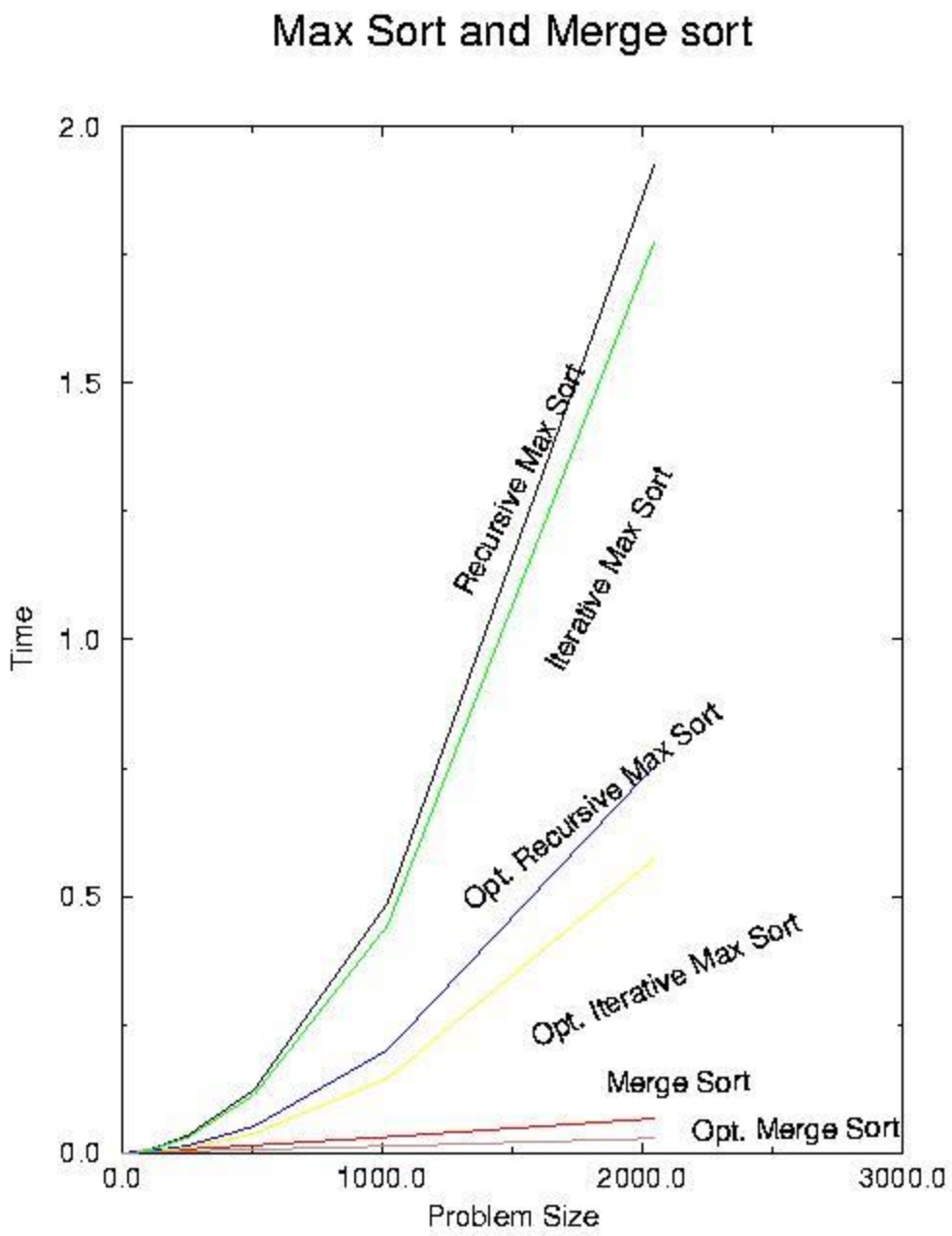
It's sometimes possible to go beyond what is first offered by the Centre for Accessible Learning.

CSC 225:


Induction
Big Oh notation
State problem
Evaluate effectiveness
Choose algorithm and data structures
Algorithm Design Process
Correctness
Develop pseudocode
Write program
Recursion
Graph Theory
Searching
Sorting
Graph Algorithms
Applications to demonstrate how the process works



Max Sort beats Merge Sort for small problem sizes.



But for bigger problems, Merge Sort is a lot faster.

We will learn paper and pencil techniques for predicting this type of behaviour.

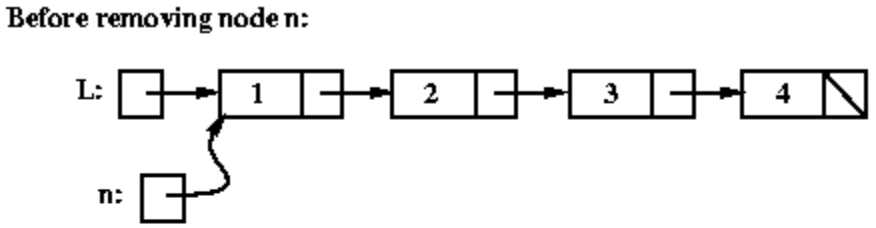
Data Structures revisited:

Stack

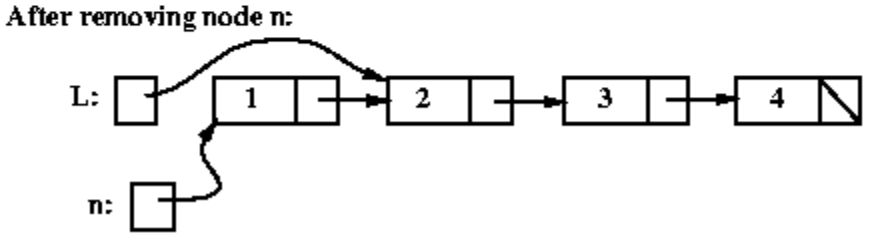
Queue

Images from: http://blog.abhisood.com/?p=69

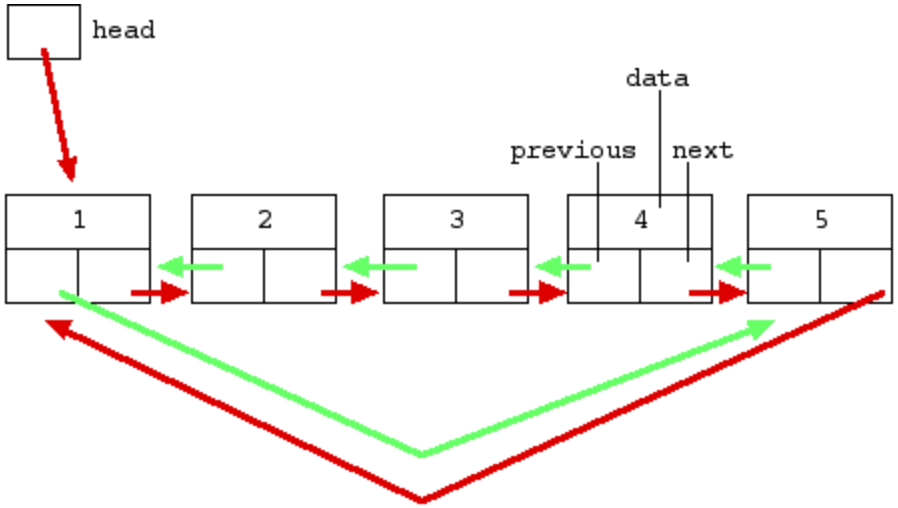
Other data structures:



Linked list



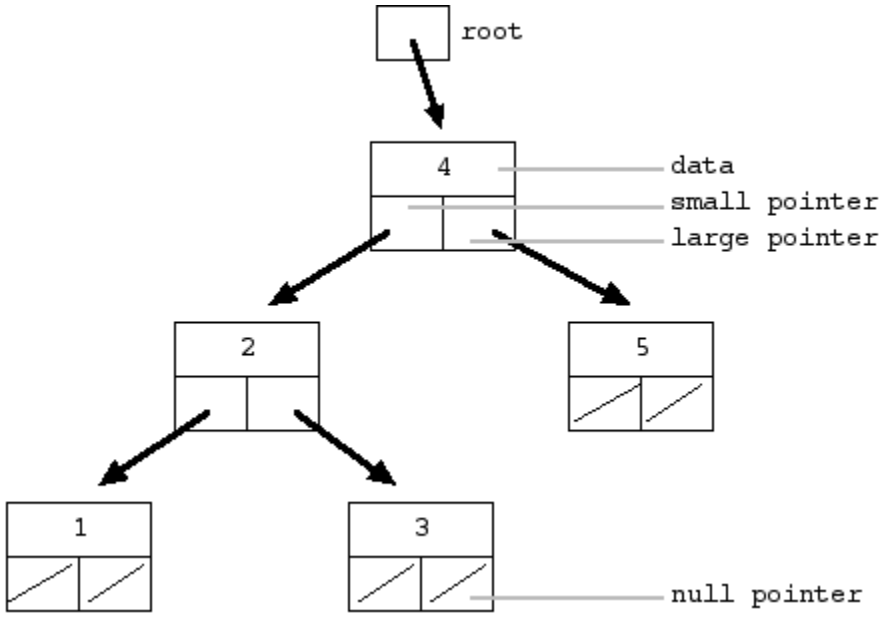
Circular doubly linked list:



http://www.itu.dk/research/algorithms/Kurser/SoegeProjekt/2003MAJ/mirrors/wiscdocs/notes/LINKED-LIST.html

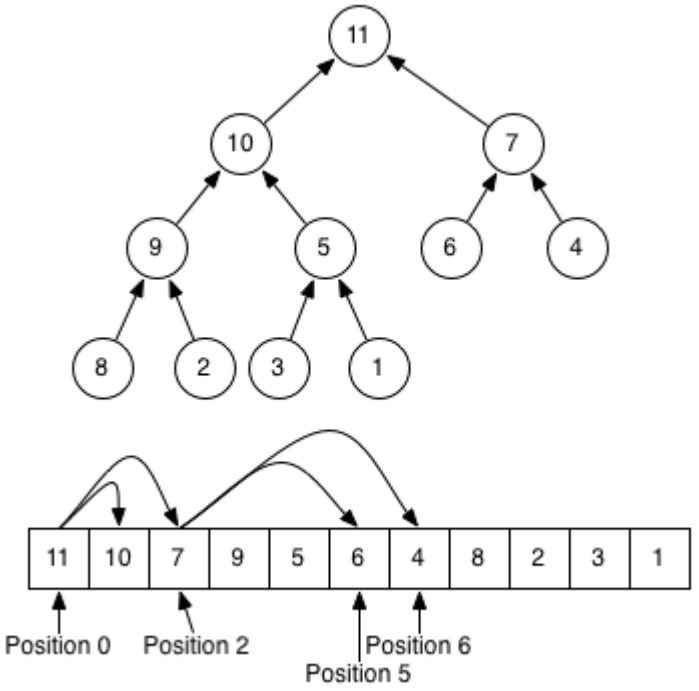
http://cslibrary.stanford.edu/109/TreeListRecursion.html

Ordered binary trees:

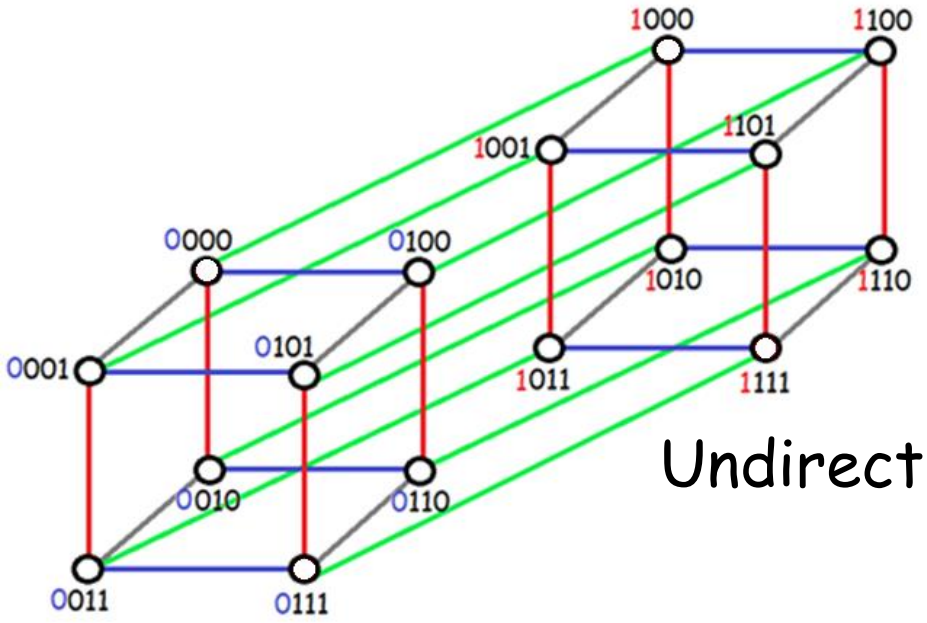


http://cslibrary.stanford.edu/109/TreeListRecursion.html

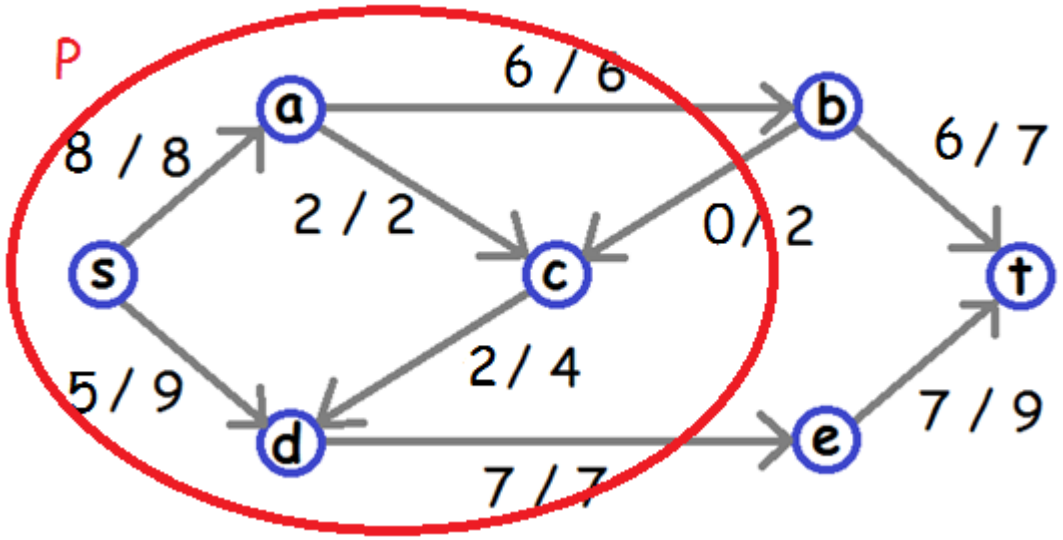
Heaps:



http://scienceblogs.com/goodmath/2008/04/implementing_compact_binary_he.php



Undirected graphs



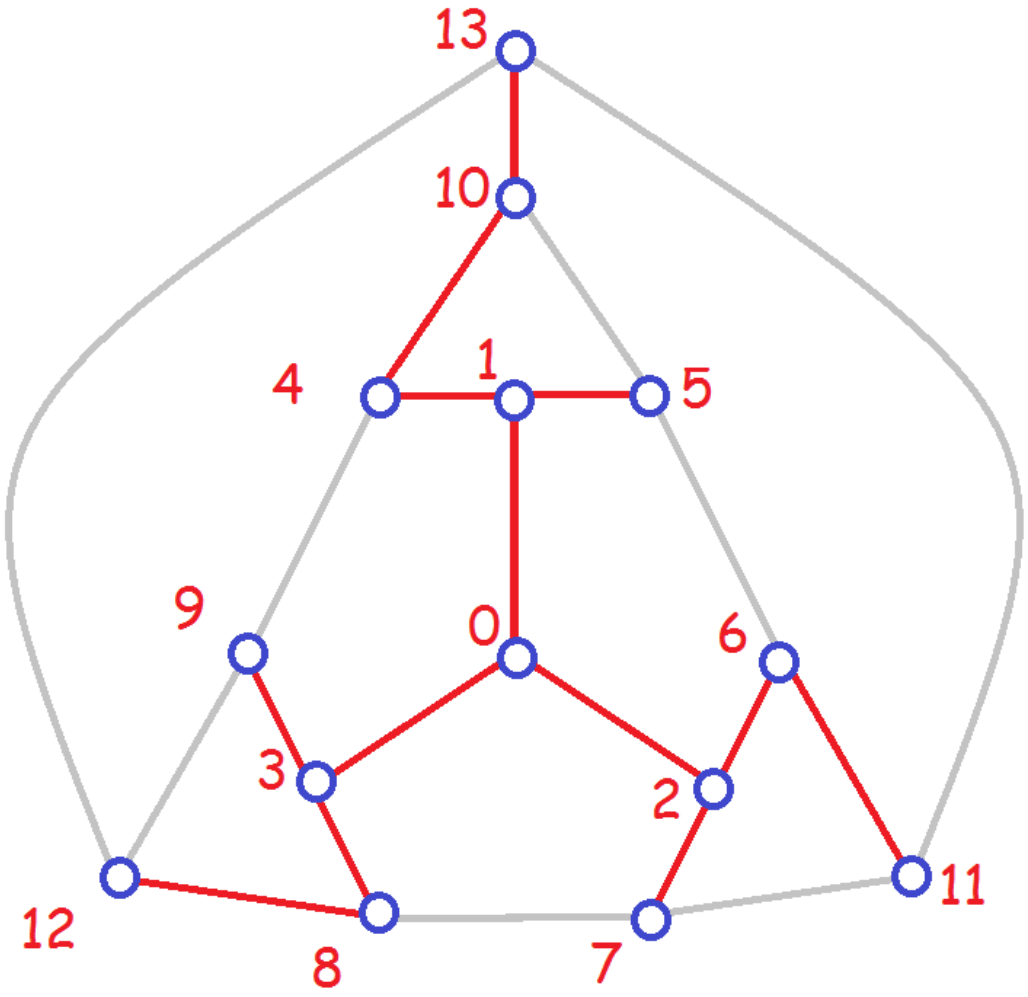
Directed Graphs

Some applications modelled by graphs:

Graph	*Nodes*	*Edges*
transportation	street intersections	highways
communication	computers	fiber optic cables
World Wide Web	web pages	hyperlinks
social	people	relationships
food web	species	predator-prey
software systems	functions	function calls
scheduling	tasks	precedence constraints
circuits	gates	wires

Table taken from Kevin Wayne.

Graph traversals: such as Breadth First Search and Depth First Search.



The red edges represent a BFS tree.

CSC 225:

The goal is to provide you with the tools to make informed decisions of algorithms and data structures for solving a problem.

What is wrong with my induction proof?

In a drunken haze I decided that the solution to the recurrence $T(1)=1$, $T(n)= 1 + T(n-1)$ is

$1 + 2 + 3 + \ldots + n$.

Theorem: The solution to the recurrence is $n(n+1)/2$.

Proof. [Basis] $T(1)=1$ and $1 *(1+1)/2 = 1$ as required.

[Induction step] Assume that $1 + 2 + \ldots + n-1 + n = n(n+1)/2$.

We want to prove that $1 + 2 + \ldots + n-1 + n + (n+1) = (n+1)(n+2)/2 = (n^2 +3n +2)/2$.

By induction, $1 + 2 + \ldots + n= n(n+1)/2$.

So $1 + 2 + .. + n + (n+1)= n(n+1)/2 + (n+1)$.

Simplifying: $(n^2 + n + 2n + 2)/2= (n^2 +3n +2)/2$ as required.