

THE

FIREWALL

 $\neg (P = NP) ?$

The claim to an age-old question that could revolutionize computing sparks speculation in the CoC community.

A WHITE AND GOLDEN HOMECOMING

VARIOUS MEMBERS OF THE COMMUNITY SHARE THEIR EXPERIENCES

OVERLOADING VIA ACADEMIC DARWINISM

BE THE FITTEST DURING THE MAD SCRAMBLE OF REGISTRATION

CONTENTS

OCTOBER 2010

4 LINUX, FROM START TO FINISH
Matthew Amidon

6 ACADEMIC DARWINISM: OVERLOADS
Tom Pilsch

8 PROGRAMMINGPUZZLER()
the Programming Team

10 (TO $P=NP$) OR \neg (TO $P=NP$)
Michael Riley

12 GRACE HOPPER CONFERENCE 2010
Jamie and Jessica Crabb

14 UPCOMINGEVENTS
the FIREWALL



COVER BY JEANIE CHOI

Checking the solution is relatively simple...
but can finding the solution be just as easy?

THE FIREWALL TEAM

PRESIDENT

Karthik Narayan

CHANCELLOR

LVL 9999

EDITOR-IN-CHIEF

Aurel Lazar

MINSTREL

LVL 9001

CHIEF COPY EDITOR

Cristina Gonzalez

COUNTRESS

LVL 3141.59

CHIEF LAYOUT EDITOR

Jeanie Choi

SORCERESS

LVL 900

LEAD WEB DEVELOPER

Andrew Panfel

MASTER CHIEF

LVL 4242

WRITERS

Jamie Crabb
Jessica Crabb
Michael Riley
Klee Simmond
Matthew Amidon

COPY-EDITING TEAM

Cristina Gonzalez
Aurel Lazar
Amanda O'Neal
Nicholas Smith

LAYOUT TEAM

Jeanie Choi
Jamie Crabb
Jessica Crabb
Taylor Kelly
Courtney Smith

WEB DEVELOPMENT TEAM

Andrew Panfel
Taylor Kelly
Courtney Smith

COMMUNITY CONTRIBUTORS

Tom Pilsch

THE FIREWALL TURNS

TWO

Hey there, CoC!

First of all, I'd like to wish FIREWALL a very happy second birthday! Personally, it has been really fun to watch FIREWALL evolve as an organization, when Ajai Karthikeyan and Mansi Sharma first put things together. The first issue, a black and white newspaper, has evolved into a nice color magazine, thanks to our dedicated staff members. Meanwhile, it's been a really busy past month, hasn't it? Midterms, Grace Hopper, first place at homecoming ... where do we begin? Meanwhile, behind stages, FIREWALL has worked really hard to produce this better-than-ever issue!

Our feature piece for this issue is about the famous P versus NP problem. Hanging around Georgia Tech and especially the CoC, chances are that you've been in a conversation or a class where P versus NP was brought up. Were you twiddling your thumbs about, looking around, feeling nonplussed... or should I say, NPed? Then read up on this introduction to P versus NP to get up to speed (no pun intended).

Did you read an interesting article that you'd like to hear more about? Let us know at feedback@gtfirewall.com!

If you'd like to become more involved with FIREWALL, whether it's with writing a single article or it's with joining the organization, please send us an email at recruiting@gtfirewall.com.

From the FIREWALL team, hope you guys enjoy this issue!
Karthik Narayan
Chancellor of FIREWALL

COME HELP SPREAD THE FIRE

You can contribute to the College of Computing Student Community!



If you are interested, contact us at
recruiting@gtfirewall.com

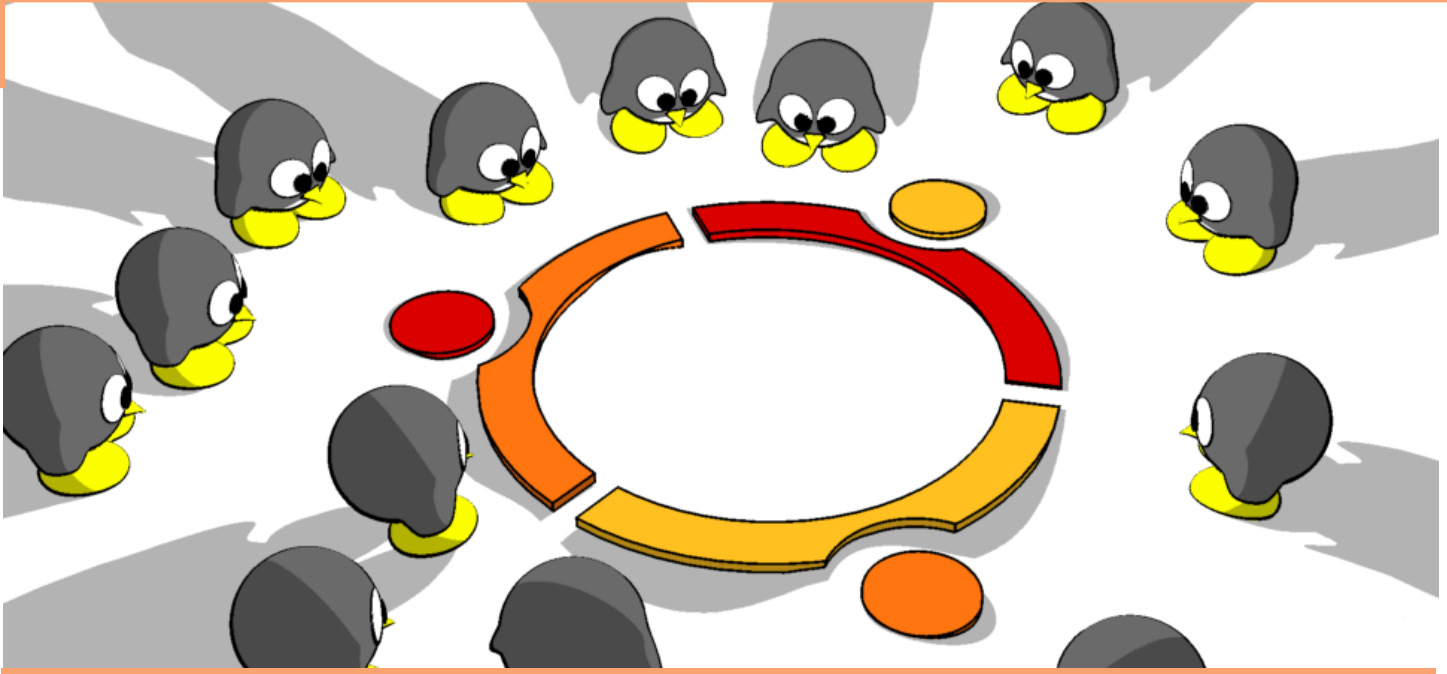
www.GTFIREWALL.COM

Writing
Layout Editing
Web Development
Illustrating
Art/Graphic Design
Photography
Sponsorship positions
Marketing/Publicity positions
Printing and Distribution

Experience is welcome but by no means necessary!

Linux, from Start to Finish

by Matthew Amidon



These days, Linux is a viable operating system for getting work done. Be it a programming assignment or that paper due tomorrow, there are free programs to help you complete any assignment.

The first thing that must be done before you can actually use Linux is install a distribution of it; I will use Ubuntu, a beginner-oriented distribution, for this article. The easiest way to install Ubuntu merely requires a 2 GB thumb drive and a free hour. Simply go to unetbootin.sourceforge.net and download the windows client. Then run it, select the Ubuntu distribution and the latest live version (10.04_live). Next, select your thumb drive and click OK. Unetbootin will then download the image file, copy it to the thumb drive, and make the drive bootable.

“unetbootin.sourceforge.net”

Once that is done, restart your computer and boot from the thumb drive. If all goes well, you will boot to an Ubuntu desktop. However, the installation is not yet done. You'll have to actually install Ubuntu to

your hard drive. To do this, click on the “install to hard drive” short cut on your desktop. Ubuntu will ask you a few questions, including user names and partition set up. Once you pick these, Ubuntu will then install itself in about half an hour.

Once Ubuntu finishes installing and you get to the desktop, there are a few changes to note before

The screenshot shows the 'Install' window for Ubuntu. The 'Who are you?' section contains the following fields and text:

- What is your name?** (Text box with 'Softpedia' entered)
- What name do you want to use to log in?** (Text box with 'softpedia' entered)
- If more than one person will use this computer, you can set up multiple accounts after installation.*
- Choose a password to keep your account safe.** (Two password fields, both with dots)
- Enter the same password twice, so that it can be checked for typing errors.*
- What is the name of this computer?** (Text box with 'softpedia-desktop' entered)
- This name will be used if you make the computer visible to others on a network.*
- ☐ Automatically sign in
- Progress indicator: Step 5 of 7
- Buttons: Quit, Back, Forward

One of the configuration steps: setting up name, login, and password.

proceeding. First, the Linux file system is somewhat different than Windows. There are no lettered drives, C, D. There is only the 'root' folder, denoted by '/'. Take the path '/home/users/you'; the equivalent windows path would be 'C:\home\users\you'. Also, executable files are not indicated by their extensions alone (.bin, .sh, etc.), they all must have the executable permission set. In Ubuntu, this permission is found under a file's properties. So, if you download an executable file and can't run it, be sure to make sure it has been marked as executable.

So, now that you've started running Ubuntu, you'll probably have run into some of the more obvious differences between Windows and Ubuntu. For starters, the equivalent of the start menu, my documents, and the control panel are now all on the left side of the upper panel. They are, respectively, applications, places, and system. On windows, if you wanted to install a random program to do something, you would go to Google. In Ubuntu, all you have to do is go to System > Administration > Synaptic Package Manager. Programs are distributed in what are called packages, which are indexed by third party servers and exposed to your synaptic client. All you have to do is search for a program, select it for installation, and then hit apply. Synaptic will then download the package, any of it's dependents, and install it to your system. It is incredibly easy, and there are over 30,000 packages listed as I write this.

What if you run into a problem? For example, your wireless doesn't work. Unlike Windows, Linux has a truly thriving community built up which is eager to help newcomers resolve their issues. You can either google

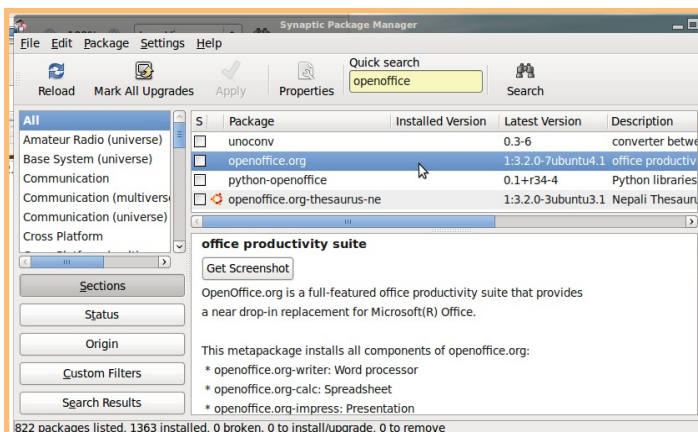
a description of your problem, or search various forums such as www.ubuntuforums.org for a solution. In the unlikely event that there isn't a forum post detailing how to solve your problem, you will most likely not need to wait more than a day or two for a correct answer.

Now that you've downloaded and resolved any issues with Ubuntu, you may want to personalize it a bit. Linux has a very open ended window management system, which allows your bland default installation to get more eye candy than even the most expensive version of Windows. Simply install your video driver, open a terminal, and type the following:

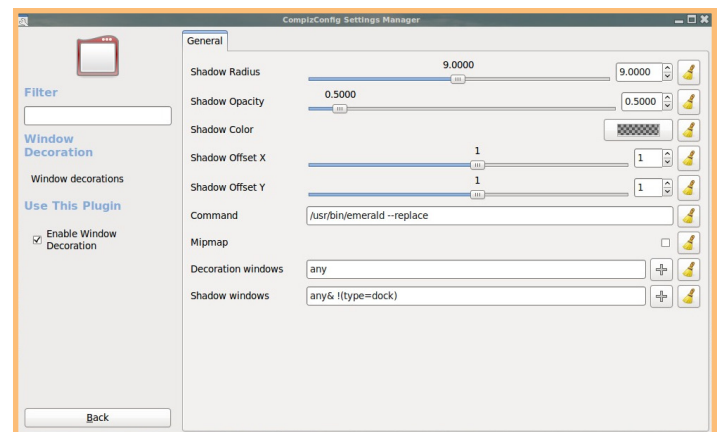
```
sudo apt-get install compiz compizconfig-
settings-manager emerald
```

This is basically the command line equivalent of synaptic which will download and install Compiz Fusion, a window manager, and Emerald, a window decorator. Then go to System > Preferences > CompizConfig Settings Manager and check the following boxes: desktop cube, rotate cube, and window decorator. Then select the window decorator (not the check box) and alter the command property from whatever it is now to '/usr/bin/emerald -replace'. Now you will have basic eye candy enabled. You can find easy to install themes at various sites like compiz-themes.org.

By now, you should have a fully functioning and rather pretty installation of Ubuntu. Hopefully, you will enjoy using Linux as much as the rest of us.



The Synaptic Package Manager.



CompizConfig Settings

Academic Darwinism

Registration 101: Overloads

MEMO FOR:
Georgia Tech Students

FROM:
The Student Services Team

by Tom Pilsch

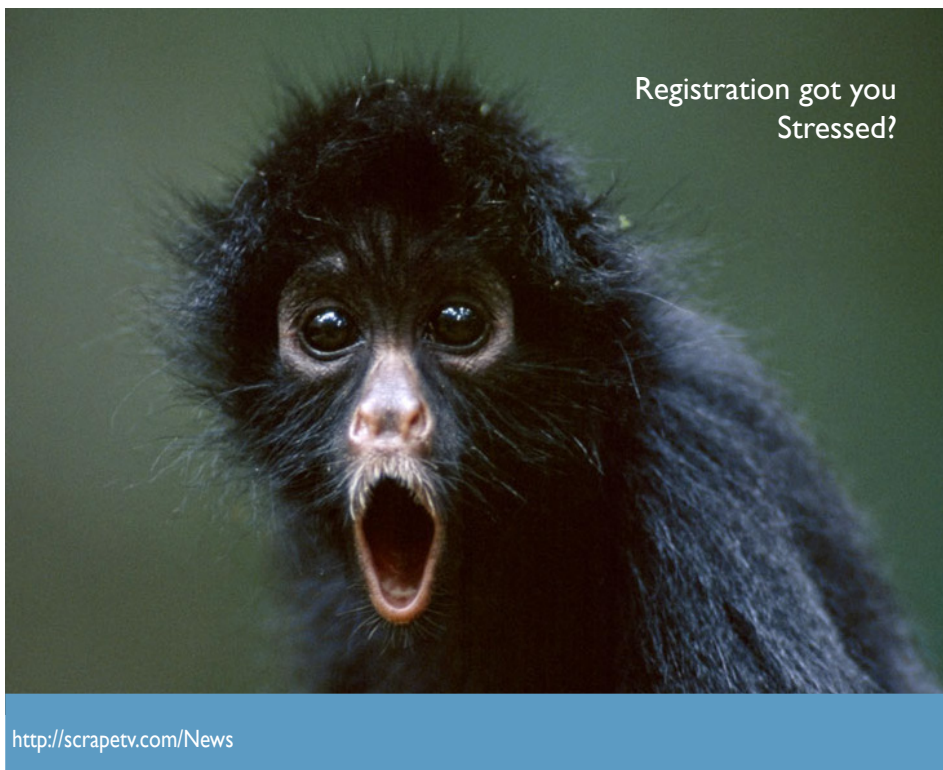
It seems that just as you get into the groove of a new semester and are starting to feel comfortable with at least some of your courses, it's times to start planning for the NEXT semester ... a life-time away! Well, Spring semester starts in less than three months, and registration is even closer – **November 1st-19th** – and that will be here in a heartbeat.

It is time to start thinking about the courses you want to take next semester. If you are serious about graduating someday, you will come in to see one of us about your options. You might even re-read and heed the advice we gave you last spring (see “*Academic Darwinism*”, FIREWALL, Feb 2010) about registering as soon as your time ticket opens. These are the things that will put you ahead of the competition in the great registration lottery.

Yet despite all your best efforts, you still might have difficulty getting the class you need at the time you want and will have to go to Oscar and apply for an overload. As daunting as this may seem to you, there are some things you can do to improve your chances of hitting the jackpot and getting that precious last seat in the class you must have. Here is our advice to you from the ghosts of registrations past.

If you are a graduating senior and apply for an overload during Phase II, the first question we are going to ask you is why you didn't register as soon as your time ticket opened in Phase I. Like figuring skating, we will grade your response on both technical and artistic merit.

The stories you may have heard about graduating senior priority in getting into any class you need are just that – urban legends. Your time ticket is your priority. If you have the hours to graduate, you likely will have one of the earliest time tickets. Use that priority or risk losing your seat to an underclassman. If you come to us at the end of Phase II for a seat in a required course that physically is full, you better start





getting your story together to tell your parents why you will be around for yet another semester. There is not much we can do for you at that point.

If you are not a senior, chances are you are not a degree candidate. Please don't claim that you are. It definitely will not help you chances to get an overload.

The worksheet sheet you receive from your advisor is not an "academic contract". That term is reserved for those returning from drop, and we know who they are. You may think the worksheet is a binding agreement, but it is just advice on what you need to progress toward graduation. It is up to you to register as soon as you are able. Your overload request comes with a time stamp, and we do compare this with you time ticket and when

the class filled.

If we tell you that all overloads are processed via OSCAR, we actually mean it. The constant emails, office visits, and phone calls from you will not change our minds, but you may remain on them for some time to come.

No, your professor cannot grant you an overload. Please read the previous paragraph again.

Spring 2011 Registration: November 1st - 19th

If you don't believe what we tell you and decide to escalate your issue to the highest levels of the Institute, please have the foresight to include your GD ID# and the course(s) you are having difficulty with. The people on The Hill will just send your e-mail back to us for resolution. We are up to our armpits in alligators during registration, and if we have to do a lot of digging to find who you are, we just might not get to your problem as quickly as you would like.

Remember that having to request an overload is like an encounter with the law. It is there to help you, but is best avoided. And you can avoid that pain by planning ahead and registering early.

If you find yourself in the position where you need to ask the system for help via an overload, remember the advice your parents gave you if you ever get stopped by the police: Stay Cool! You can help us do that by not generating any more heat than necessary. We really do have your best interests at heart.

One last thing:

Remember the First Law of Academic Darwinism: ■

The Seats Go to the Swift



ProgrammingPuzzler()

by **The Programming Team**

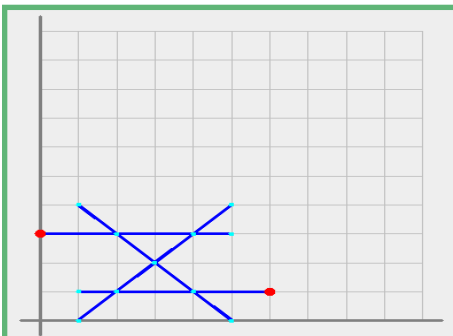
A deranged algorithms professor has devised a terrible final exam: he throws his students into a strange maze formed entirely of linear and circular paths, with line segment endpoints and object intersections forming the junctions of the maze. The professor gives his students a map of the maze and a fixed amount of time to find the exit before he floods the maze with xerobiton particles, causing anyone still in the maze to be immediately inverted at the quantum level. Students who escape pass the course; those who don't are trapped forever in a parallel universe where the grass is blue and the sky is green.

The entrance and the exit are always at a junction as defined above. Knowing that clever ACM programming students will always follow the shortest possible path between two junctions, he chooses the entrance and exit junctions so that the distance that they have to travel is as far as possible. That is, he examines all pairs of junctions that have a path between them, and selects a pair of junctions whose shortest path distance is the longest possible for the maze (which he rebuilds every semester, of course, as the motivation to cheat on this exam is very high).

The joy he derives from quantumly inverting the majority of his students is marred by the tedium of computing the length of the longest of the shortest paths (he needs this to know to decide how much time to put

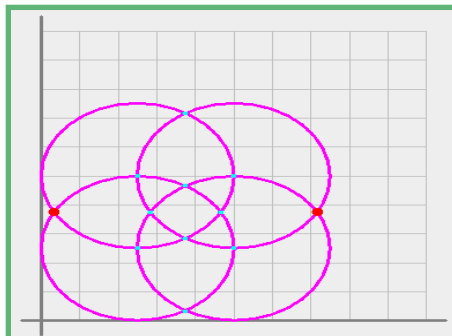
on the clock), so he wants you to write a program to do it for him. He already has a program that generates the mazes, essentially just a random collection of line segments and circles. Your job is to take that collection of line segments and circles, determine the shortest paths between all the distinct pairs of junctions, and report the length of the longest one.

The input to your program is the output of the program that generates his mazes. That program was written by another student, much like yourself, and it meets a few of the professor's specifications: 1) No endpoint of a line segment will lie on a circle; 2) No line segment will intersect a circle at a tangent; 3) If two circles intersect, they intersect at exactly two distinct points; 4) Every maze contains at least two junctions; that is, a minimum maze is either a single line segment, or two circles that intersect. There is, however, one bug in the program. (He would like to have it fixed, but unfortunately the student who wrote the code never gave him the source, and is now forever trapped in a parallel universe.) That bug is that the maze is not always entirely connected. There might be line segments or circles, or both, off by themselves that intersect nothing, or even little "submazes" composed of intersecting line segments and circles that as a whole are not connected to the rest of the maze. The professor insists that your solution account for this! The length that you report must be for a path between connected junctions!



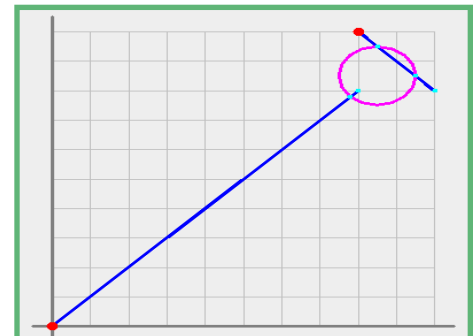
Line segments only. The large dots are the junction pair whose shortest path is the longest possible.

(See File1.in)



Note that in this case there is also another pair of junctions with the same length longest possible shortest path.

(See File2.in)



The line segments are connected by a circle, allowing for a longer shortest path.

(See File4.in)

details

Input

An input test case is a collection of line segments and circles. A line segment is specified as "L X1 Y1 X2 Y2" where "L" is a literal character, and (X1,Y1) and (X2,Y2) are the line segment endpoints. A circle is specified by "C X Y R" where "C" is a literal character, (X,Y) is the center of the circle, and R is its radius. All input values are integers, and line segment and circle objects are entirely contained in the first quadrant within the box defined by (0,0) at the lower left and (100,100) at the upper right. Each test case will consist of from 1 to 20 objects, terminated by a line containing only a single asterisk. Following the final test case, a line containing only a single asterisk marks the end of the input.

Output

For each input maze, output "Case N: ", where N is the input case number starting at one (1), followed by the length, rounded to one decimal, of the longest possible shortest path between a pair of connected junctions.

examples

File1.in

```
L 10 0 50 40
L 10 40 50 0
L 10 10 60 10
L 0 30 50 30
```

Outputs: 68.3

File2.in

```
C 25 25 25
C 50 25 25
C 25 50 25
C 50 50 25
```

Outputs: 78.5

File3.in

```
L 0 0 80 80
L 80 100 100 80
```

Outputs: 113.1

File4.in

```
L 0 0 80 80
L 80 100 100 80
C 85 85 10
```

Outputs: 140.8

Please send solutions to [w](#). Good luck!

If you are an algorithms enthusiast, you have doubtlessly heard about the latest proof from Vinay Deolalikar of HP labs which claimed $P \neq NP$. It took the academic community less than 2 months to find fundamental flaws within the proof, but many expressed hope that this is a good first draft to a final solution.

w

But what is P and NP? And why is it important for computing?

P and NP both describe a general set of problems that describe what a Deterministic Turing Machine (a computer) can and can not accomplish. The set 'P' defines all problems that can be solved in a set polynomial time. Algorithms that are solved in $O(n)$, $O(n^2)$ are all in P. An example of a problem within P is, determine the greatest common denominator of integers a and b. There is a simple algorithm we can follow to determine the factors of our variables, and the number of computations we must run is bounded by the size of the integers, so clearly a computer can solve this problem.



Vinay Deolalikar, Principal Research Scientist
Source: HP Labs (hpl.hp.com)

(To $P=NP$) OR \neg (To $P=NP$)

by Michael Riley

'NP' problems however, have a much greater range of complexity, and cannot be solved by a Turing machine. They generally have an exponential complexity, such as $O(2^n)$, or $O(n^n)$. A famous example of an NP problem is the traveling salesman problem. The problem is easy to explain: Given a set of cities, come up with the shortest tour around each city. Interestingly, the traveling salesman problem could be solved if you had many computers all calculating the shortest tour from a given path, but you would need a theoretically infinite number of machine to calculate the exact answer. So the problem falls in the realm of 'NP'. We have algorithms that can come up with "good" answers, but they are not always correct. An interesting side effect of NP problems is that verifying the correct answer can actually fall within the realm of P. Problems that are easy to

verify, but hard to derive is the basis for algorithms that ensure secure encoding of messages, and the birth of computer security as we know it today.

One of the earliest encryption algorithms is the Rivest-Shamir-Adleman algorithm (RSA), which uses encryption and decryption keys for sending secure messages. If we have person A, Alice, how do send a message to person B, Bob, without having person C, Charlie intercept and read the message? Each person has a public encryption key, which everyone can use to encrypt a message to that user. Everyone also has a private decryption key that each specific person can use to decrypt messages sent to them. So all Alice has to do is use Bob's encryption key on her message, and send it to Bob. If Charlie reads it, then all he sees is garbage. But Bob has his

private decryption key, only he can use it to get back the original message. As long as it's really hard to derive the decryption key given the encryption key, the system is secure.

Both the encryption and decryption keys are just integer values. The process for generating the keys (D and E respectively), relies on the NP problem of prime factorization. It is really hard to derive prime factorization for really big composite numbers. So as long as you cannot efficiently factorize, then RSA is secure. In order to generate the keys, you must:

Pick 2 unique prime numbers: p and q . For this example, I will use 7 and 13, but the algorithm recommends picking 100 digit decimal numbers.

Calculate $n = p \times q$, and calculate $\phi = (p-1)(q-1)$. Our n is equal to 91, our ϕ is equal to 72.

Choose an integer e between 1 and ϕ exclusive. This is our encryption key value. The algorithm recommends picking large values for e , but our range is relatively small anyways. I go with 5.

Calculate another integer d such that $(d \times e) \% \phi = 1$. You can easily write this as $\gcd(d, \phi) = 1$, and use a reversed Euclid's algorithm to solve for d . Clearly, your decryption key is depended on your encryption key. Our lowest possible d for this problem is 29. We can verify this by the algorithm above: $(29 \times 5) \% 72 = 1$.

Now that we have both of the keys how do we actually encode a message? If someone wanted to send us the message "hello mike", they would first have to encoded the message into numbers, and block each section into "message chunk". For this example I'll use a small, simple alphabet. "A" = 1, "B" = 2, "C" = 3, etc. We'll also just make whitespace = 99 just for this example "hello mike", when blocking into 2 digit blocks, would look like: 08 05 12 12 15 99 13 09 11 05. Then to encode, you take each message block, raise to a power of e (your encryption key), and mod by n . In algorithm form, it reads as $M^e \pmod n$. Using our e from above, the first block becomes $(8^5) \% 91 = 8$. The next block is $(5^5) \% 91 = 31$. The entire encrypted message reads as 08 29 00 00 63 27 61 09 59 29. To decrypt, you run the same algorithm, with your decryption key instead. If you run the algorithm, you can see that the original message returns: 08 05 12 12 15 00 13 09 11 05.

In this example, our n was comparably small, so our message chunks were one character each. However, this means that each character hashed out to a certain value, and all Charlie has to do is key each character with an encrypted value. In practice, your end up grouping many characters at a time, so that keying each possible value quickly becomes infeasible. An encoding method that more complex than simple substitution is more than sufficiently encoded. Of course, this encryption method is also broken the moment that Charlie is able to figure out d . Fortunately, in order to figure out d , Charlie must figure out p and q . Since factorization is an NP problem, choosing a really large p and q means that it would take the best known computer algorithm years to calculate. The record for highest number factored on a computer was 768 digits long, and most keys today are at least 1024 digits.

But what if Vinay Deolalikar is wrong, and $P=NP$? Then suddenly all of these problems thought to be impossible for computers to solve efficiently would now, by definition, have a solution. Factorizing a large number would suddenly be solvable in polynomial time, and RSA encryption would no longer be secure. In fact, most modern day cryptography would no longer work, as they rely on the fact that some problems are easy to verify, but hard to compute. Fortunately, mathematicians are skeptical that a $P = NP$ proof can even exist, and most believe that $P \neq NP$ is much more likely to be true. ■

Err, what is this $P=NP$?

- P represents all problems whose solution can be found in polynomial time.
- NP represents all problems whose solutions can be verified in polynomial time.
- Polynomial time just means that the time to find the solution has a polynomial relationship with the size of the problem. For instance, a particularly terrible sorting algorithm may take n^2 seconds to sort an arbitrary list of names, where n is the number of names in the list. This is a second-order polynomial of the form $(a+b) \times (n+c) \times (n^2)$, where $a = b = 0$, and $c = 1$.
- And finally, does $P=NP$?

Grace Hopper Conference Celebrates Women in Computing

by Jamie Crabb and Jessica Crabb

The Grace Hopper Celebration of Women in Computing (GHC), named in honor of one of the very first computer programmers, was held from 9/28 -10/2 in Atlanta this year. It was the largest GHC so far with 2,147 attendees representing 29 countries and 280 schools. Georgia Tech had the largest group with 69 women attending. The conference provided a unique opportunity for the women at Georgia Tech as well as the rest of the women in the technical field to meet with companies, have their résumés critiqued, network with each other, display their research, learn about new opportunities, and get inspired by the successful women already working in the field.

The career fair allowed for students to meet with 75 notable companies, including Adobe, Amazon, Apple, AT&T, Cisco, Comcast, Facebook, Google, HP, IBM, Intel, Microsoft, and Yahoo!. The poster session allowed students to highlight their research; Georgia Tech presented two posters. "How to Attract Students to Computing?" presented by Barbara Ericson featured ways to attract students (especially females and underrepresented minorities) from the fourth to twelfth grades to computing. Her poster was a big hit with attendees in part because it showcased the adorable Pleo robotic dinosaurs.



Victoria Ayo posing holding a Pleo, with Barbara Ericson and her poster in the background.

Inspiration came to the audience of technical women in the form of panels, sessions, and keynote addresses. A record 630 speakers attended this year's GHC. One notable keynote speaker was Carol Bartz, CEO of Yahoo! and one of two female CEOs of Fortune 500 companies. She spoke of the importance of embracing change, standing up for yourself and standing up to the guys. In her most memorable quote she said "It's very easy to get put into roles...to not let our stars shine. Sometimes you have to be [aggressive]. Stand up for yourself. You have to manage your own career." After her speech, awards were given to inspiring women in technology - those who have used technology to impact the lives of other women in their communities and around the world.



Computer Engineer Barbie

The popular doll Barbie will be adding Computer Engineer to her long resume thanks to online voters. She will be wearing a binary code tee, comfortable wedges and a hot pink laptop. This will hopefully be a positive influence for young girls interested in computing.

"Sometimes you have to be [aggressive]. Stand up for yourself. You have to manage your own career."

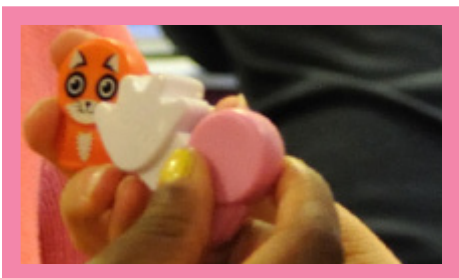


Carol Bartz, CEO of Yahoo!, inserting some sass into her speech.

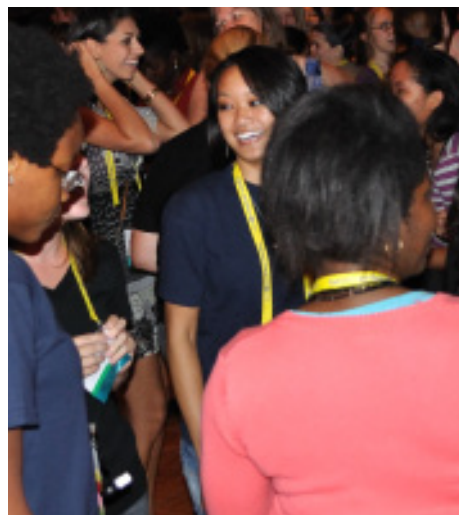


Poken

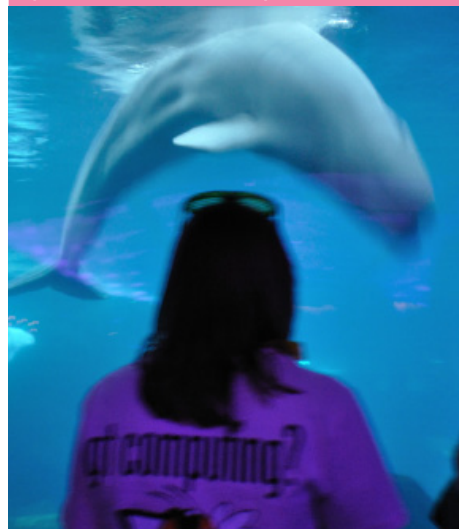
Poken is a social business card that automatically transfers your information to a new contact. It features a USB, a hand for virtual handshakes and an online location for contact information. Students were given pokenSparks free. The pokenSpark showcases a cute design with easy functionality and was the primary way to connect with other attendees and presenters at GHC.



When 2,000 women come together for one cause, there is sure to be a good time had by all. One of those good times was the Georgia Tech sponsored DJ Dance Party, which allowed the women in technology to let loose, have fun, and get to know the other attendees. Another memorable highlight was GHC Sponsor Night at the Georgia Aquarium sponsored by Microsoft and Google. (Don't worry; there weren't any



W@CC members at the Georgia Tech sponsored Dance Party.



Georgia Tech's GHC shirt in front of a Beluga Whale at the Aquarium party.

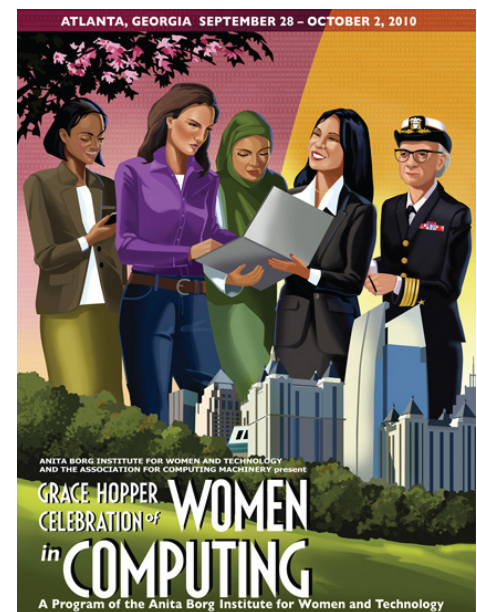


Women enjoying dinner at the dance party at the Georgia Aquarium, sponsored by Google and Microsoft.

Bing vs. Google fights.)

The music, food, free t-shirts, views of Beluga whales and chance to meet smart, successful women definitely made for an unforgettable combination.

The Grace Hopper Conference was a truly unique and amazing experience for women in computing, and reportedly the largest gathering to date of women in technology. Every year the Grace Hopper Celebration of Women in Computing grows as the inclusion of more women in the computing field becomes a more widespread goal. This winter a sister Grace Hopper Conference will be held in India to grow its community of female computer scientists. Next year's conference will be held in November in Portland, Oregon. ■



The official poster for the 2010 GHC exemplifies this year's motto: **"Collaborating across Boundaries".**

All pictures by Jamie Crabb and Jessica Crabb, except for the following:
Barbie: mattel.com
Poken: pokenlive.com

UPCOMING EVENTS

Hungry Hungry Hackers

Saturday, October 23rd
10 AM - Midnight
GTRI Conference Center

October						
S	M	T	W	T	F	S
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

W@CC Disney Karaoke Night

Friday, October 22nd
5 PM
Klaus III 6W

Enjoy an evening singing Disney songs and meeting new people!

ESP Smash Brothers Tournament

Saturday, October 30th
5-11 PM
Piedmont room, Student Success Center

Entrants must bring own controllers.



EVERY Monday 6 - 7 pm FIRE GENERA



November

S	M	T	W	T	F	S
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

Microsoft College Puzzle Challenge

Saturday, November 6th
12 PM - 11 PM

Save the world from the Apocalypse on a Saturday!
Register online at www.collegepuzzlechallenge.com

Turkey Ball

Saturday, November 20th
8 PM - 1 AM

Klaus III 6

Conjured up by Minorities@CC in collaboration with Hispanics@CC, Women@CC, and other student organizations, the Turkey Ball will serve as a fundraising event for unprivileged families. Want to dance, eat delicious food, have an awesome evening, and help donate to charities on November 20th? Simply bring canned food (at least two) to be admitted.

Upsilon Pi Epsilon Interview Fair

Tuesday, November 9th
11-2PM
Klaus III 6

Come talk to others who have done internships in various companies (and get swag!)





WWW.GTFIREWALL.COM