



summerprogram in  
mathematical  
problemsolving

A PROGRAM OF THE AOPS FOUNDATION

## SPMPS Alumni Newsletter

March, 2015

### The Google Issue



*Google's logo in the lobby of their NYC offices.*

On February 20, SPMPS 2014 alumni visited Google and got to hear from Googlers about math and the work that they do.

First, the SPMPSErs split up into small groups to have lunch in many of the cafeterias across the huge building. After lunch, we got back together to hear talks from the four Googlers about how they use math in their work. (You can read all about it in this issue.) Then we heard from a panel of Googlers about the work they do and how they got there, and finally we got a tour of

the building and got to enjoy the many microkitchens with their snacks (and in one place, smoothies!).

What did Googlers have to tell us? Well, we learned a lot of math and how it's used at Google. But the big message we kept getting was to follow your passions. We heard about Googlers who loved math, or programming, or something. In middle school and high school, they spent a lot of time on whatever they loved, and it paid off. It made them happy, and colleges and employers saw that they were the kind

of people who learn because they're passionate about it. They weren't just about grades or SAT scores. The people who made it to Google got out there and did something! Many of them did math contests, or independent research, or summer programs, or computer programming on their own. They made it happen.

So the advice they had was simple. Follow your passions. Do it in the real world. Create something. Do research. Go somewhere and learn something big. It pays off!

## Dan's Challenge Problem



Send in a solution to have it printed and win a free book!

Split the grid into pieces so that (a) all pieces are made up of the same number of squares, and (b) all pieces have the same sum of the numbers inside them. (The pieces cannot overlap, and they must each contain only whole squares.)

				6	5	6
4			2		4	
	3		3		4	
				4		
		4				3
	4			4		4
1	1	1				

Source: USAMTS, a free mail-in math contest

**How to submit:** Send your answer to [spmps@artofproblemsolving.org](mailto:spmps@artofproblemsolving.org), or mail it to:  
SPMPS  
PO Box 4499  
New York, NY 10163



## Using Statistics to Understand Computer Code

Googler Justin Venezuela, who works on the Google Docs team, talked about an interesting problem he ran into. Google Docs is a massive codebase, and thousands of programmers have worked on it, each making little pieces. As a result, sometimes something gets programmed into Docs that is unnecessary.

While working to clean out unnecessary parts of the code for Docs to make it faster and more efficient, he discovered that every document he looked at had a certain setting marked "no". He didn't know what this setting did, and he had no way to find out; some other programmer had inserted it, probably many years ago. Since it was marked "no" everywhere he saw, he thought it was probably "no" for every document and he could safely delete the setting. But if he was wrong, then important information would be lost, and Google Docs might break!

There are so many files on Google Docs that it would take too long even to have a computer program check to see if this setting was really "no" for every document. So instead, Justin decided to do a statistical test. He would test enough documents that he could be 99.99% sure that at least 99.999999% of documents had this setting marked "no".

This is something called a confidence interval. It's also used to do polls for elections, which is why you might hear about the "margin of error" for a poll. It's also used when doing scientific measurements. Someone who tries to measure the size of an atom wouldn't be able to get it exactly, but they'd be 95% sure that it's in a certain confidence interval.

Anyway, Justin did the calculation, and figured out that to get the confidence interval he needed, he had to randomly sample 9.2 million documents and check that this setting was "no". This was a number he could handle (there are a lot more Google Docs than this!), so Justin wrote a computer program to do it. When the program verified that the setting was "no" on all 9.2 million documents, he could safely delete the setting off of every Google document, knowing that it was very unlikely that it would cause any problems. This saved Google a small amount of computational resources for each document - which adds up to a lot when you think of how many documents there are!



SPMPS 2014 alumni listen to the panel discussion at Google. (See first page.)



## Sorting

Googler Jessie Newman works on Google Drive as well. In Drive, you can sort your file names in alphabetical order. Jessie taught us how to do this, and then how to do it faster.

The first method Jessie taught us is called bubble sort. Say that you have

a "word", or a sequence of letters, like GOOGLE. To put the letters in alphabetical order, bubble sort goes through and switches the first pair of letters that are in the wrong order (to the right, they're written in red).

The problem is that this takes a long time! It's even worse if the letters start in reverse order. Imagine you started with ZYXWVUTS. First, it would take 1 flip to make the Y the first thing in the list. Then it would take 2 flips to move the X to the first spot. Then it would take 3 flips to move the W to the first spot. Now it's 4 flips to move the V, 5 flips to move the U, 6 flips to move the T, and 7 flips to move the S, for a total of

$$1 + 2 + 3 + 4 + 5 + 6 + 7 = 28 \text{ flips!}$$

In fact, you might remember that to add the first n digits, you do  $n(n+1)/2$ . So doing this for a word with k letters if it starts in reverse order takes  $(k-1)k/2$  flips, which is a lot!

But, if you split the word in half first, then it goes faster. For example, if you sort ZYXW it only takes 6 flips, and sorting VUTS also only takes 6 flips. Then after they're sorted merging them is pretty easy, since to merge WXYZ and STUV you just look at the first letter, take the earlier one, and keep doing that until you're out of letters. In fact, we could have saved even more flips by splitting ZYXW into ZY and WX (only one flip each!) and splitting VUTZ into VU and TZ (also one flip each), then merging, and merging again. This way to sort is much faster, and it's called **merge sort**.

### Sorting GOOGLE:

GOOGLE  
GOGOLE  
GGOOLE  
GG~~O~~LOE  
GGLO~~O~~E  
GGLO~~E~~O  
GGLE~~O~~O  
GG~~E~~L~~O~~O  
G~~E~~GL~~O~~O  
EGGL~~O~~O

### Sorting ZYXWVUTS:

ZYXWVUTS  
YZXWVUTS  
YXZ~~W~~VUTS  
XYZ~~Z~~VUTS  
XYW~~Z~~VUTS  
XWY~~Z~~VUTS  
WXY~~Z~~VUTS

## Using Someone Else's Sort Function

When someone programs something they use a lot (like sorting), they save the code for others to use in the future, making what's called a library. So you don't have to write a way to sort yourself: you can load someone else's library and use theirs, which they've made really fast by using merge sort.

Googler Paul Cheong talked about a problem he had to solve, where he had a library that could sort some numbers, but he wanted the numbers to be sorted in descending order, and the library sorted in ascending order. So he wanted it to take a list like 5, 18, 3, 29, 30 and turn it into 30, 29, 18, 5, 3. Instead, his library would turn it into 3, 5, 18, 29, 30.

How could he use the library he had, which was really efficient, to do what he wanted it to do?

Rashik Ahmed (SPMPS Siena '14) saw a solution right away: take the negative of each number. If you sort -5, -18, -3, -29, -30 in ascending order, you get -30, -29, -18, -5, -3. Then just take the negative again and you get them in descending order: 30, 29, 18, 5, 3. Unfortunately, in Paul's case he couldn't do this because the numbers were stored as unsigned integers, which means that they couldn't be negative. (Turning them into signed integers would have taken more computation when you have a really big list.)

Instead, Paul found a bigger integer than all of them and subtracted each from his big integer. So if he subtracts them all from 40, he'd get 35, 22, 37, 11, 10. If he sorts these numbers in ascending order, he gets 10, 11, 22, 35, 37. Now if he subtracts all of these numbers from 40, he'd get 30, 29, 18, 5, 3. That's the list he wanted in descending order!

Of course, that's because

$$40 - (40 - x) = 40 - 40 + x = x.$$

It's just the distributive law!

So Paul found a way to use his library and still do the calculation quickly. (In fact, when he actually did it he just subtracted the numbers from MAXINT, the biggest integer that you can use with an unsigned integer, so he didn't even have to worry about finding a number bigger than every number in his list.)



# Last Month's Challenge Problem Solutions

Congratulations to Afia Boake (SPMPS '14 Siena) who solved last month's challenge problems!

The first problem was to find the last digit of:

$$5 \times 10 \times 15 \times \dots \times 2015$$

The problem was actually a typo! It was supposed to be  $5 \times 15 \times 25 \times \dots \times 2015$ . But Afia answered the problem I stated correctly: the last digit is 0, because we multiply by 10.

The second problem asked which of the following numbers is a perfect square:

$\frac{14!15!}{2}$	$\frac{15!16!}{2}$	$\frac{16!17!}{2}$
$\frac{17!18!}{2}$	$\frac{18!19!}{2}$	$\frac{19!20!}{2}$

$(17! \times 18!)/2$  is a perfect square, because it is equal to  $(17! \times 17!) \times 9$ , which is a product of perfect squares. The others are not: for example,  $(15! 16!)/2$  is  $(15! \times 15!) \times 8$ . But taking the square root gives  $15!$  times  $\sqrt{8}$ , and since 8 is not a perfect square, neither is this number.

## Quick Updates

**Summer Programs:** Many programs are filling up, so let us know if you want to find a good program for you.

**Weekly Office Hours:** Thursdays, 3pm-8pm, 1150 6th Ave., 6th floor. Drop in any time!

**Purple Comet Math Team:** You should have gotten a letter from us. If you want to join our Purple Comet math team, be there on April 11 for our team practice! Contact us if you have any questions.

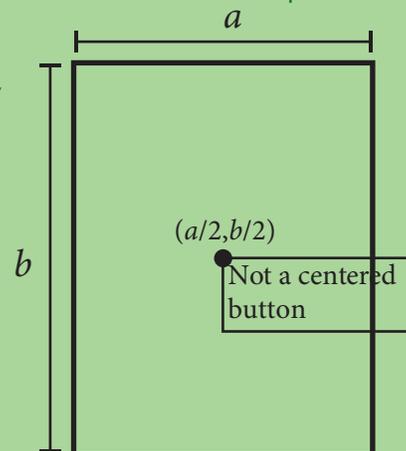
**SAT Preparation:** 11th graders, we've started having some people come in every Thursday to work on the SATs. Join us to prepare!

## Using Geometry for an iPhone App

Googler Kevin Veloso works on making the Google Drive app for iPhones, iPads, and other iOS devices. His mathematical problem was how to place a button on the screen. Say that he wants to put a button right in the center of your iPhone. The way to tell the iPhone where the button should go is to specify the coordinates of the top-left corner.

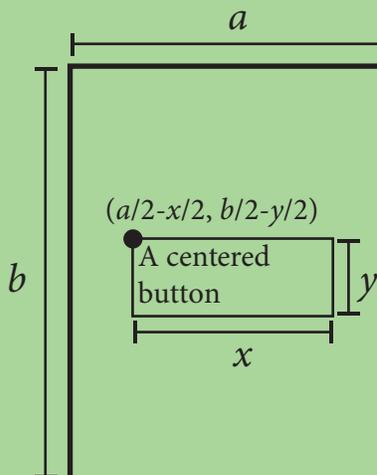


Coordinates are given in terms of pixels. A pixel is a single dot on a screen. So if the screen is  $a$  pixels wide and  $b$  pixels tall, and the button is  $x$  pixels wide and  $y$  pixels tall, where do you tell the iPhone to put its top left corner?



What you don't do is tell it to put the corner in the very center of the screen! The center of the screen has coordinates  $(a/2, b/2)$ . But if you put the top-left corner of the button there, then the button isn't centered! You need the center of the button to be in the center of the screen, not the top-left corner of the button to be in the center of the screen. (See the first picture.)

So how do you do it? Well, you need to move the button to the left by half its width. So instead of making the first coordinate  $a/2$ , you make it  $a/2 - x/2$ . You also need to move the button up by half its height, so you make the second coordinate  $b/2 - y/2$ . (See the second picture.) Hence, your final coordinates are:



$$((a - x)/2, (b - y)/2)$$