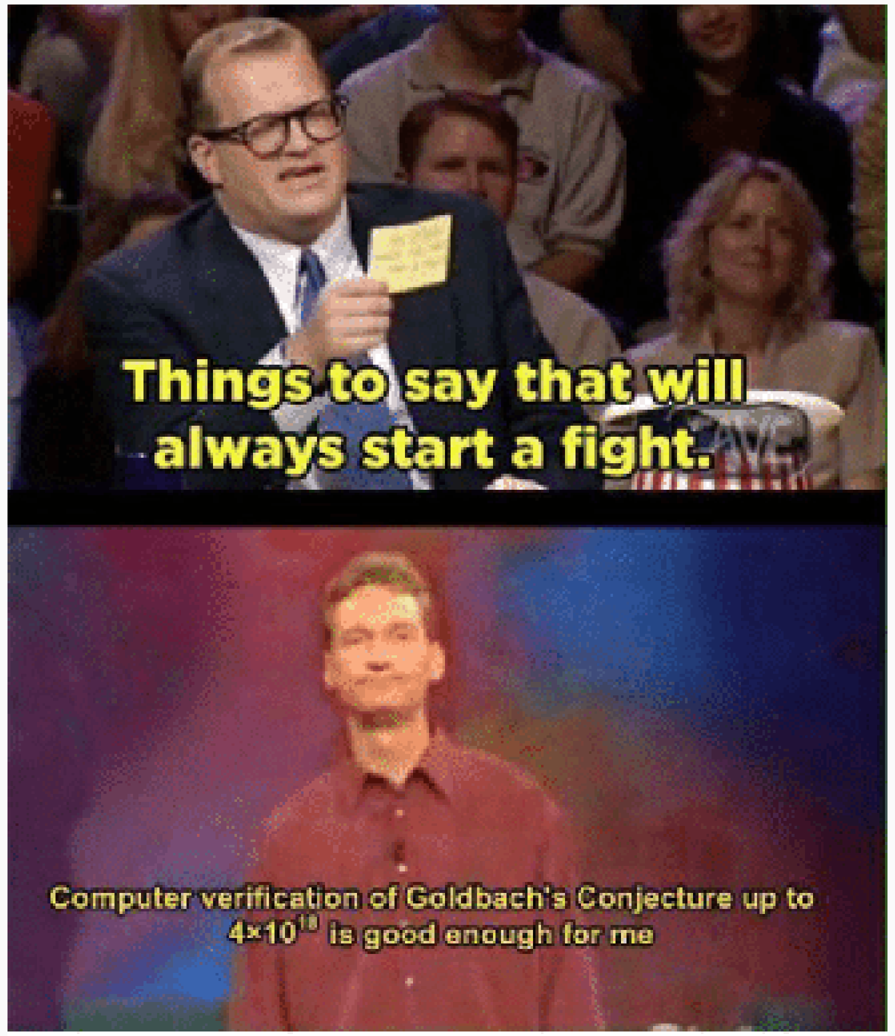
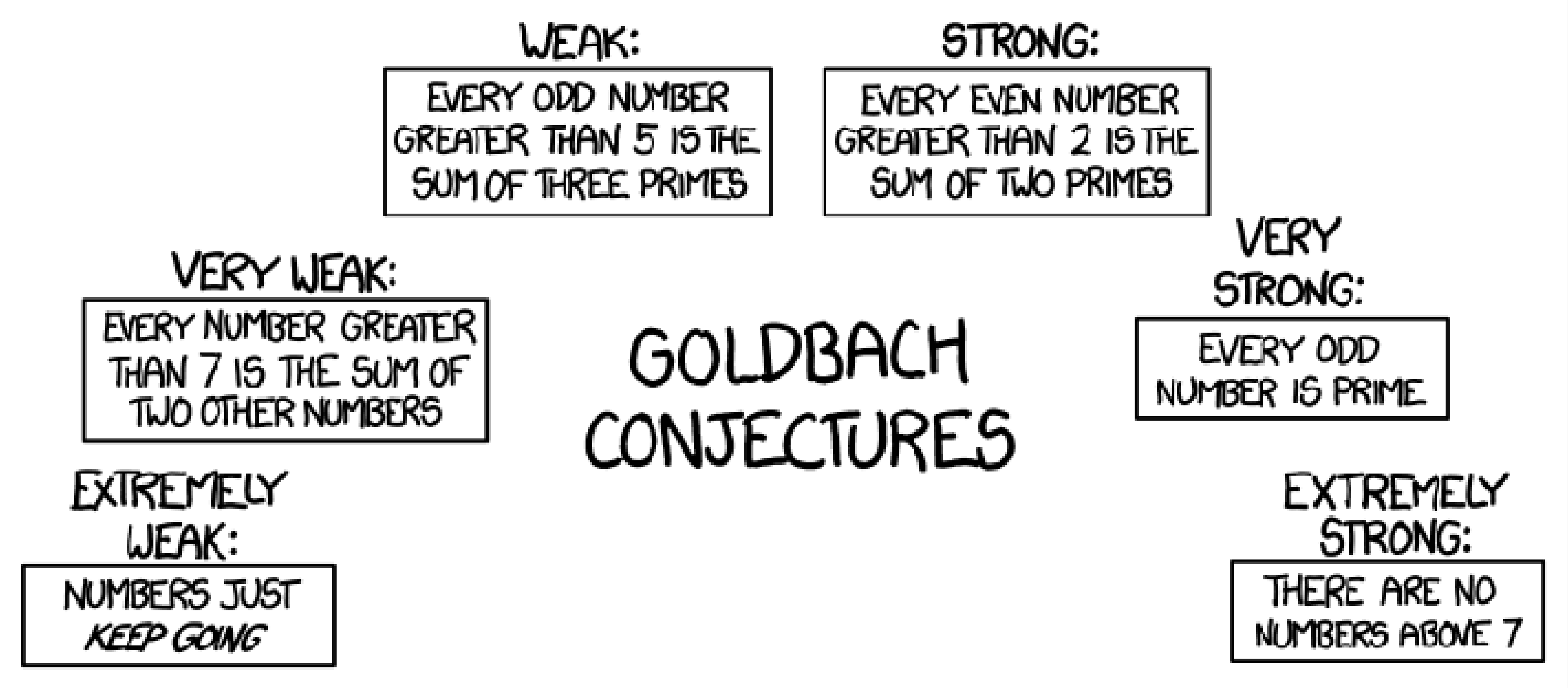
Instructions: We'll use the same whiteboard process until, as a whole class, we decide to use some other process.

Note: The links in the PDF are not showing up properly in Google Drive view (some TeX setting, I'm sure). Click on the word and it will still take you to the web page. Or download the file and you can see the links.

And I present (not my work): Goldbach memes and a Goldbach xkcd comic (Other famous math xkcd comics: https://xkcd.com/435/ https://xkcd.com/263/ https://xkcd.com/410/ Hover on the comic to read the caption.)





1

1. Apply the Sieve of Eratosthenes to find all the primes less than 100.

Should have stopped at 7 instead of 13...

	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	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

-Miah

2 a (Explain process.)

2. **a.** Find four other twin prime pairs (i.e. 8 total primes). For this one, don't look up online but come up with a method of your own.

$$(41, 43), (59, 61), (71, 73), (101, 103), (107, 109), (137, 139), (137, 139), (149, 151).$$

```
def primes(min=2,max=100):
p=[2,3,5,7]
if min>=2 and max>2:
    for i in range(min,max+1,1):
        if i%2!=0 and i%3!=0 and i%5!=0 and i%7!=0:
            p+=[i]
    return p
else:
    print("Invalid value(s) chosen")
    return

ps=primes()

def twins(ps):
    tp=[]
    for i in range(0,len(ps)-1,1):
        if ps[i+1]-ps[i]==2:
            tp+=[(ps[i],ps[i+1])]
        else: pass
    return tp
```

Cy

2 b

b. Did you notice any properties that twin primes should have from your investigation?

I noticed that for twin primes p_i, p_{i+1} that $p_{i+1}^2 - p_i^2 = 4p_i + 4$.

The number between the twin primes must even and divisible by 3 so that the other primes are 2 and 1 mod 3. This means they "straddle" multiples of six

5 is the only prime that appears in two twin prime pairs.

Cy

3 a
(Explain process.)

4=2+2, 6=3+3, 8=3+5, 10=5+5, 12=5+7, 14=7+7, 16=5+11, 18=5+13, 20=7+13, 22=5+17, 24=5+19, 26=3+23, 28=23+5, 30=23+7

-Maggie

3 b (Briefly explain code/algorithm, if not pseudocode.)

```
>>> import sympy
>>> primes = list(sympy.primerange(0, 2000))
>>> for n in range(4, 2000, 2):
...     i, j = 0, len(primes) - 1
...     while primes[i] + primes[j] != n:
...             if primes[i] + primes[j] < n:
...                     i += 1
...             elif primes[i] + primes[j] > n:
...                     j -= 1
...     print(f"{n} = {primes[i]} + {primes[j]}")
...
```

"primes" is a list of all primes less than 2000

i and j are indices for the prime list. They always start at opposite ends of the list.

For each even number 2 < n < 2000, start by checking if the smallest prime in the list and the largest add to n.

If they don't and their sum is too small, increase the smaller prime to the next smallest prime.

If they don't and their sum is too large, decrease the larger prime to the next largest prime.

Once the two primes sum to n, print n along with those two primes.

Could we also do something like this: Make a list of all primes. Then starting with the first term p[1] in the list, check if n-p[1] is in the list of primes. Then move to the next term, etc.

Because it has to search every time, I think this will be less efficient. But with this method, we can find all instances of creating n as a sum of two primes. Not sure if Nick's method can do that.

-Nick

3 c

The strong conjecture says that for all even N greater then 2, there exist primes p1 and p2 such that

p1 + p2 = N

That means that if we add three, an odd prime number, we can get

p1 + p2 + 3 = N + 3

Since N > 2 and even, then N + 3 > 5 and is odd. We would then have written all of the odds as the sum of three primes, implying the weak conjecture

Cian

More number theory news: https://www.quantamagazine.org/james-maynard-solves-the-hardest-easy-math-problems-20200701/ (Just fresh out of the oven.)

Last page