The Number 2024

This week's Fiddler on the Proof (22 December 2023) asks:

> There are several rectangular prisms with integer edge lengths that have an internal diagonal of length 2024. What is the greatest volume among these prisms?

The length of the internal diagonal d of a rectangular prism having dimensions $a \times b \times c$ is given by

$$a^2 + b^2 + c^2 = d^2$$

We are told that $d = 2024$, so the problem is to find integer values for a, b, and c that satisfy

$$a^2 + b^2 + c^2 = 2024^2$$

I used a computer to look for solutions. The following lines of code (written in Swift) do that.

```
for a in 1...2024
    {
    for b in a...2024
        {
        for c in b...2024
            {
            if a*a + b*b + c*c != 2024*2024 { continue }
            let volume = a * b * c
            print( volume, a, b, c )
            }
        }
    }
```

There is one optimization we could make here. Instead of looping over all the values of c, we could compute c directly as

$$c = \sqrt{2024^2 - a^2 - b^2}$$

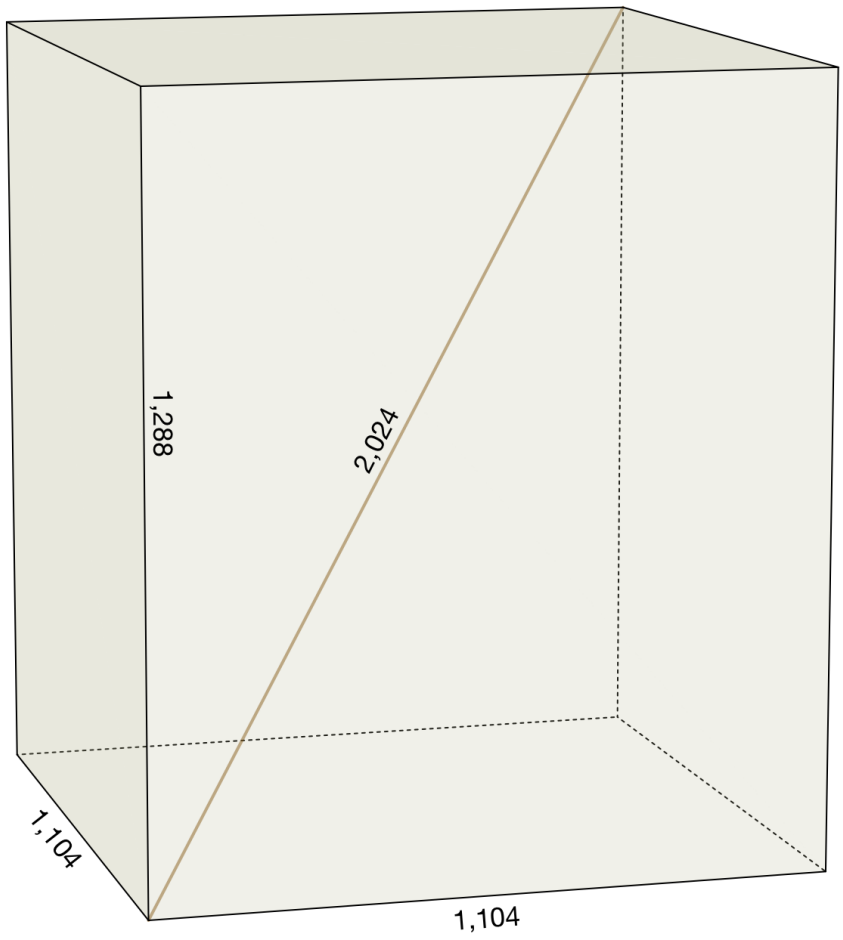
then observe whether c is an integer or not. This avoids the inner loop in our code and makes it run significantly faster. But the original code is so simple, and runs in just a couple of seconds, that I stayed with it.

There are 41 rectangular prisms with integer dimensions and an interior diagonal of 2024:

	Volume				Diagonal
	$a \cdot b \cdot c$	a	b	c	$\sqrt{a^2 + b^2 + c^2}$
1.	21,676,032	64	168	2016	2024
2.	29,417,472	96	152	2016	2024
3.	29,491,200	24	640	1920	2024
4.	46,006,272	24	1152	1664	2024
5.	130,056,192	64	1344	1512	2024
6.	152,690,688	96	872	1824	2024
7.	161,021,952	96	936	1792	2024
8.	177,979,392	96	1088	1704	2024
9.	239,542,272	152	864	1824	2024
10.	258,048,000	224	600	1920	2024
11.	269,862,912	264	528	1936	2024
12.	285,401,088	224	672	1896	2024
13.	299,778,048	192	856	1824	2024

14.	309,657,600	280	576	1920	2024
15.	339,591,168	168	1344	1504	2024
16.	384,565,248	192	1304	1536	2024
17.	390,758,400	424	480	1920	2024
18.	429,907,968	224	1176	1632	2024
19.	515,192,832	264	1232	1584	2024
20.	519,782,400	480	576	1880	2024
21.	525,312,000	360	800	1824	2024
22.	593,952,768	424	768	1824	2024
23.	672,786,432	368	1104	1656	2024
24.	700,416,000	600	640	1824	2024
25.	713,318,400	360	1376	1440	2024
26.	767,950,848	576	744	1792	2024
27.	812,777,472	424	1248	1536	2024
28.	910,835,712	576	928	1704	2024
29.	956,786,688	528	1144	1584	2024
30.	1,046,642,688	672	936	1664	2024
31.	1,059,028,992	576	1216	1512	2024
32.	1,084,243,968	576	1368	1376	2024
33.	1,188,569,088	672	1176	1504	2024
34.	1,243,348,992	744	1088	1536	2024
35.	1,334,181,888	928	936	1536	2024
36.	1,345,978,368	768	1304	1344	2024
37.	1,354,752,000	800	1176	1440	2024
38.	1,435,779,072	856	1248	1344	2024
39.	1,437,253,632	864	1216	1368	2024
40.	1,483,702,272	936	1152	1376	2024
41.	1,569,835,008	1104	1104	1288	2024

The one with the largest volume is illustrated below.



Extra Credit

The Collatz Conjecture asserts that if you start with *any* positive integer and repeatedly apply a certain operation, the resulting sequence of numbers will always reach the number 1. The operation is: For even numbers divide by 2; for odd numbers, multiply by 3 and add 1.

For example, if you start with 11, the sequence goes: 11 → 34 → 17 → 52 → 26 → 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1.

What is the *smallest* starting number for which 2024 appears somewhere in its Collatz sequence?

Here I turned to the computer again. The following function computes the Collatz numbers generated from a given starting number.

```
func collatzSet( startingAt:Int ) -> Set<Int>
    {
    var i = startingAt
    var set : Set = [ i ]
    while i > 1
        {
        i = i.isMultiple( of:2 ) ? i / 2 : i * 3 + 1
        set U= i
        }
    return set
    }
```

The heart of this function is the line

```
i = i.isMultiple( of:2 ) ? i / 2 : i * 3 + 1
```

which implements the *Collatz operation*: "For even numbers divide by 2; for odd numbers, multiply by 3 and add 1."

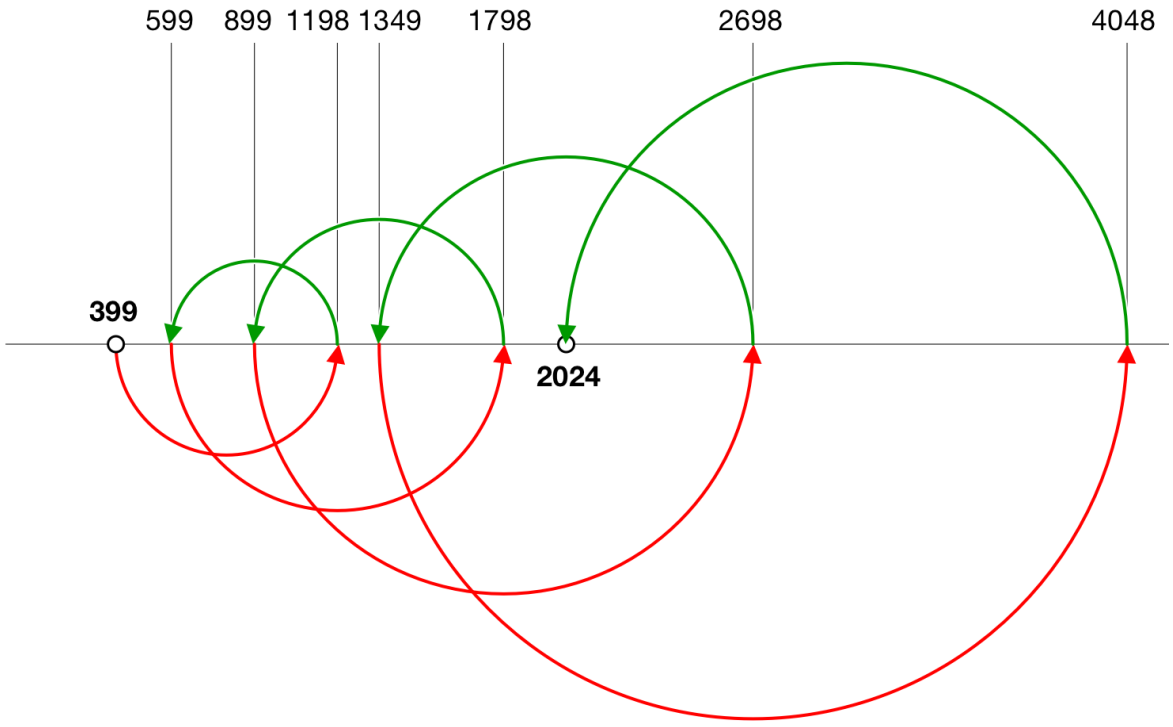
Now we can go through the starting numbers and look for a Collatz sequence that contains 2024:

```
for i in 1...2024
    {
    if collatzSet( startingAt:i ).contains( 2024 )
        {
        print( "The Collatz sequence starting at \(i) contains 2024." )
        break
        }
    }
```

Running this code informs us that:

```
The Collatz sequence starting at 399 contains 2024.
```

The value 399 reaches 2024 in 8 steps, as shown here:


599
899
1198
1349
1798
2698
4048
399
2024