064

July 23, 2018

- The continued fractions are generated based on a recurrence relation. a_n the value of interest, b_n, the numerator, and c_n.
- Solving for the recurrence relation gives us:

$$a_n = int\left(\frac{b_{n-1}}{\sqrt{N} - c_{n-1}}\right)$$
$$b_n = \frac{N - c_{n-1}^2}{b_{n-1}}$$
$$c_n = b_n * a_n - c_{n-1}$$

- If the continued refraction has a repeat of all a,b, and c then the fraction has repeated and its period can be calculated

In [30]:
```
import time as time
```

In [65]:
```
#def SqrtPeriod(N):

def SqrtPeriod(N):

    if (N**0.5).is_integer():
        return 0
    values = [((int(N**0.5)),1,(int(N**0.5)))]

    while True:

        an = (int(values[-1][1]/(N**0.5 - values[-1][2])))
        bn = (int((N - values[-1][2]**2)/values[-1][1]))
        cn =(int(bn*an-values[-1][2]))

        try:
            return len(values) - values.index((an,bn,cn))
        except:
            values.append((an,bn,cn))
```

In [67]:
```
start = time.time()

Count = 0
```

```
        for n in range(2,10001):
            if SqrtPeriod(n)%2 == 1:
                Count = Count + 1

        print(Count,'Time Elapsed:%s'%(time.time()-start))
```

```
1322 Time Elapsed:1.32818603515625
```