Homework 3:

Buffon's needle

1.

$$P = \int_0^{t/2}\int_{arcsin(\frac{x}{l/2})}^{\pi/2} \frac{4}{t\pi} d\theta dx = \int_0^{t/2} \frac{4}{t\pi}(\pi/2 - arcsin(\frac{x}{l/2}))dx = 1 - \frac{4}{t\pi}(\frac{t}{2}arcsin(t/l) + \sqrt{(l/2)^2 - (t/2)^2}) - l/2)$$

2.

General Code for l>t and l<t two different cases:

```
import numpy as np
import matplotlib.pyplot as plt

class BuffonNeedle():
    def __init__(self,lLength,tLength,times=100):
        self.l = lLength
        self.t = tLength
        self.N = int(times)

    def run(self):
        ratio = self.l/self.t
        x = np.random.random(self.N)*self.t/2
        theta = (np.random.random(self.N))*90
        count = 0
        for i in range(0,self.N):
            test = 0.5*self.l * np.sin(math.radians(theta[i]))
            if x[i] < test :
                count +=1
        Pi = (2*self.l/self.t/(count/self.N))
        return Pi
```

When l=2, t=3 ,N = 10E5, the result is 3.1372549019607843

Part 3:

```
import numpy as np
import matplotlib.pyplot as plt

class BuffonNeedle():
    def __init__(self,lLength,tLength,times=100):
        self.l = lLength
```

```
        self.t = tLength
        self.N = int(times)

    def run(self):
        ratio = self.l/self.t
        x = np.random.random(self.N)*self.t/2
        theta = (np.random.random(self.N))*90
        count = 0
        for i in range(0,self.N):
            test = 0.5*self.l * np.sin(math.radians(theta[i]))
            if x[i] < test :
                count +=1
        Pi = ( 2*ratio-2*np.arcsin(1/ratio)-2*np.sqrt(ratio**2-1))/(count/self.N-1)
        return Pi
```

when l=3, t=2 N = 10E5, result is 3.13848721387351

Part 4:

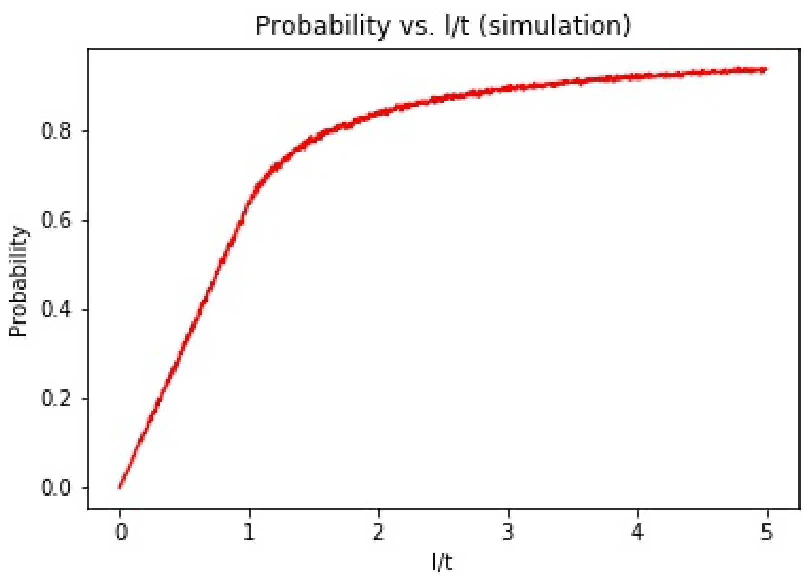

Figure 1:

Part 5:

We can find that our simulation is pretty good: they are almost the same. When I try to plot them in one pad, I find I couldn't distinguish them because they are almost coincide.

So we can say that the for this problem, MC simulation perform well enough to

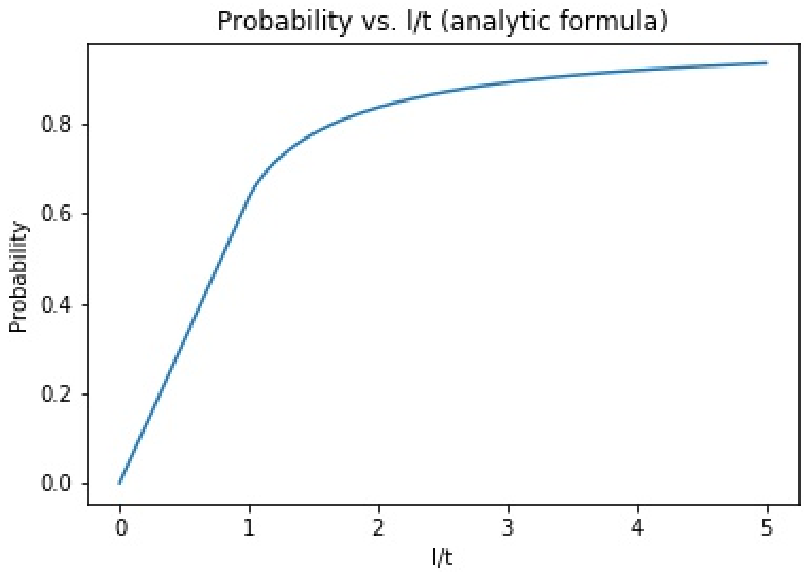

Figure 2:

solve it.