```python
from random import *
import math
import matplotlib.pyplot as plt # Import plot
import numpy as np

tries = 100000 # 10^5 tires
needleLength = 0.5 # set the needle length
ASpacing = 1 # set the spacing between the first set of parallel lines
BSpacing = 3 # set the spacing between the second set of parallel lines
topAngle = math.atan(ASpacing / BSpacing) # find the top angle in radians
CSpacing = 0.3 * math.sqrt(10) # set the spacing between the thired set of parallel lines
longrun = [] # for storing all long-run values

n = list(range(0, tries))
fig = plt.figure()

for run in range(0, 10): # Ten runs
    estimation = [] # for storing the estimations
    hits = 0 # for keeping track of the number of needles that cross a line
    for needle in range(0,tries): # one needle per try
        # Initialize the (x, y) position of one end of the needle at random
        # Assuming the needle is put uniformly across the board
        # z is the diagnol-perspective position
        x = random() * ASpacing
        y = random() * BSpacing
        z = random() * CSpacing

        # Choose the angle that the needle makes with the horizontal in radians
        # we assume it is uniformly distributed in [0,2pi] radians
        angle = random() * 2 * math.pi

        # Use the generated angle and needle length to find (x1, y1) position of the other end
        x1 = x + math.cos(angle) * needleLength
        y1 = y + math.sin(angle) * needleLength
        z1 = z + math.cos(angle - topAngle) * needleLength
        # check if the needle cross lines in set A
        if x1 > ASpacing or x1 < 0:
            hits += 1
        # then check if the needle cross lines in set B
        elif y1 > BSpacing or y1 < 0:
            hits += 1
        # finally check if the needle cross lines in set C
        elif z1 > CSpacing or z1 < 0:
            hits += 1

        # For every try, record the estimated probablity
        estimation.append(hits / (needle + 1))
    plt.plot(n, estimation, label = "#" + str(run + 1) + " run")
    longrun.append(estimation[len(estimation) - 1])
plt.legend(loc = 'best', prop={'size':8})
plt.xlabel("Iterations")
plt.ylabel("Estimated Probability")
plt.title("Probability over iterations")
fig.savefig('p1.png', dpi = fig.dpi)
plt.xlim((50000, 100000))
range_max = max(longrun)
range_min = min(longrun)
plt.ylim((range_min, range_max))
plt.title("Probability over iterations after 50000 tries")
fig.savefig('p1_variation.png', dpi = fig.dpi)

print("Range of the long-run estimation is (" + str(range_min) + ", " + str(range_max) + ").")


#============= Output=============================================================#
```

```
# Output for tries = 10^5
# [Running] python "c:\Users\Johnnia\Desktop\46\Fall 2017\Math381\HW4\tempCodeRunnerFile.py"
# Range of the long-run estimation is (0.58559, 0.59012).

# [Done] exited with code=0 in 5.58 seconds

# Output for tries = 10^6
# [Running] python "c:\Users\Johnnia\Desktop\46\Fall 2017\Math381\HW4\tempCodeRunnerFile.py"
# Range of the long-run estimation is (0.586249, 0.587688).

# [Done] exited with code=0 in 34.827 seconds
#==============================================================================================#
```