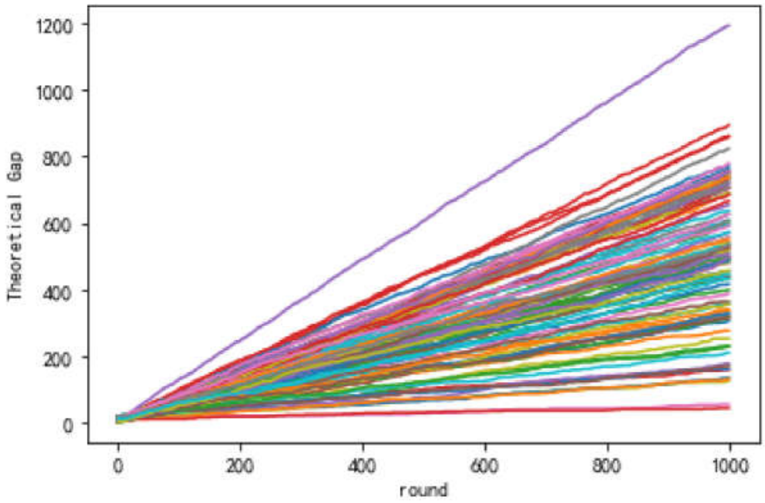
In [1]:

```
import numpy as np

# let's try 100 games each with 1000 rounds,
num_runs = 100
T = 1000

diff = np.zeros((num_runs, T))
for run in range(num_runs):
    n = np.random.randint(1,10)
    ps = np.random.random(n)
    ws = np.ones(n)
    p_g = np.random.random()
    eta = 0.25

    # your mistakes
    m = 0
    # experts mistakes
    ms = np.zeros(n, dtype=np.int32)

    for t in range(T):
        # game correct answer
        z = np.random.random() >= p_g
        # experts answers
        xs = np.random.random(n) >= ps
        # your answer
        y = np.sum(xs * ws) >= np.sum((1-xs) * ws)

        # your mistakes
        m += (y != z)
        # experts mistakes
        ms[xs != z] += 1

        # theoretical bound
        th_ub = 2 * (1+eta) * min(ms) + 2. * np.log(n) / eta
        diff[run, t] = th_ub - m

        # update weights
        ws[xs != z] = (1-eta) * ws[xs != z]


assert np.min(diff) >= 0, "Theoretical gap should be non-negative"
```

In [3]:

```
import matplotlib.pyplot as plt
%matplotlib inline

_ = plt.plot(diff.T)
plt.xlabel('round')
plt.ylabel('Theoretical Gap')
plt.show()
```


Theoretical Gap
round