2.

In [4]:

```
import numpy as np
import scipy.stats as st
import matplotlib.pyplot as plt
%matplotlib inline
from math import factorial

theta = 1
size_N = 10000

def function(sample, k):
    return (factorial(k) / np.mean(sample ** k)) ** (1. / k)

def difference (sample, k):
    y = np.array([], dtype=float)
    for q in range(1, size_N):
        y = np.append(y, abs(function(sample[:q], k) - theta))
    return y
```

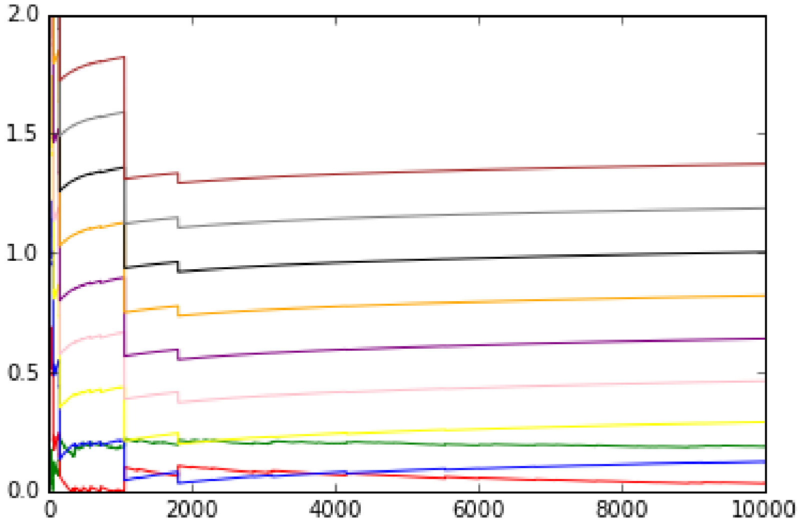
In [5]:

```
def expon_estimate (k):
    n = np.arange(1, size_N, dtype=int)

    distribution = st.expon(theta)
    sample = distribution.rvs(size = size_N)

    plt.ylim([0, 2])
    plt.plot(n, difference(sample, k * 1), 'green')
    plt.plot(n, difference(sample, k * 2), 'red')
    plt.plot(n, difference(sample, k * 3), 'blue')
    plt.plot(n, difference(sample, k * 4), 'yellow')
    plt.plot(n, difference(sample, k * 5), 'pink')
    plt.plot(n, difference(sample, k * 6), 'purple')
    plt.plot(n, difference(sample, k * 7), 'orange')
    plt.plot(n, difference(sample, k * 8), 'black')
    plt.plot(n, difference(sample, k * 9), 'grey')
    plt.plot(n, difference(sample, k * 10), 'brown')
    plt.show()
```

Построим графики зависимости модуля разности оценки $\left(\frac{k!}{\overline{X^k}}\right)^{\frac{1}{k}}$ и параметра θ для $k \in$ {5, 10, 15, 20, 25, 30, 35, 40, 45, 50}

In [7]:

```
expon_estimate(5.)
```



Лучшая оценка среди перечисленных достигается при k = 10 (красный цвет)