```
import numpy as np
          import random
          import time
          def num(p,n ill,N):
              roll = np.random.rand(N-n ill)
              return np.sum(roll<(1-(1-p)**n ill))</pre>
In [3]: %%time
          num(0.25, 250,1000)
         Wall time: 0 ns
Out[3]: 750
          def avg_num(iterations, p, n_ill,N):
              k=0
              for i in range(iterations):
                   k+=num(p, n ill,N)
              return k/iterations
          avg num(20, 0.5,1,1000)
In [8]:
Out[8]: 497.3
          def graph(iterations, n_ill,days,N, step=10):
              amount=np.empty(step)
              p=np.linspace(0.001,0.01,step)
              for i in range(step):
                   avg_vect=np.ones((iterations, days))*n_ill
                   for j in range(days):
                       for k = in  range(iterations):
                           avg vect[k, j]=round(num(p[i],int(avg vect[k, j-1]),N))
                   plt.plot(range(days), np.mean(avg_vect, axis=0), label='p='+str(p[i]) )
          %%time
          graph (1, 1, 50, 1000)
         Wall time: 35.9 ms
          600
          500
          400
          300
          200
          100
           0
                               20
                                        30
                       10
                                                         50
          graph(20,1,50,1000)
In [34]:
          600
          500
          400
          300
          200
          100
           0
               0
          def graph(iterations, n_ill,days,N, step=10):
In [44]:
              amount=np.empty(step)
              p=np.linspace(0.001,0.5,step)
              for i in range(step):
                   avg_vect=np.ones((iterations, days))*n_ill
                   for j in range(days):
                       for k in range(iterations):
                           avg vect[k, j]=round(num(p[i],int(avg vect[k, j-1]),N))
                   plt.plot(range(days), np.mean(avg vect, axis=0), label='p='+str(p[i]) )
              plt.legend()
          graph(1,1,50,1000)
In [45]:
          800
                  p=0.001
                  p=0.05644444444444444
                  600
                  p=0.167333333333333333
                  p=0.22277777777777777
                  p=0.278222222222222
          400
                  p=0.3336666666666667
                  p=0.3891111111111111
                  p=0.4445555555555554
          200
           0
               Ó
                       10
                               20
                                        30
                                                         50
In [46]:
          graph(20,1,50,1000)
                                        p=0.001
                                        800
                                        p=0.111888888888888888
                                        p=0.167333333333333333
                                        p=0.22277777777777777
          600
                                        p=0.278222222222222
                                        p=0.3336666666666667
          400
                                        p=0.3891111111111111
                                        p=0.4445555555555555
          200
           0
                       10
                               20
                                        30
                                                40
          def limited_num(iterations, n_ill, days,N):
              step = 100
              p_c=[]
              pos=0
              amount=np.empty(step)
              p=np.linspace(0,0.004,step)
              for i in range(step):
                   avg_vect=np.ones((iterations, days))*n_ill
                   for j in range(days):
                       for k in range(iterations):
                           avg_vect[k, j]=round(num(p[i],int(avg_vect[k, j-1]),N))
                   amount[i] = np.mean(avg_vect, axis=0)[-1]
              for l in range(step):
                   if amount[1]==0:
                       pos+=1
              pos-=1
              return p[pos]
                plt.plot(p, amount)
          limited_num(10, 1, 100, 1000)
         Wall time: 2.15 s
         0.00101010101010101
In [40]:
          def limited_avg_num(iterations, n_ill, days,N):
              step = 100
              amount=np.empty(step)
              p=np.linspace(0,0.004,step)
              for i in range(step):
                   avg_vect=np.ones((iterations, days))*n_ill
                   for j in range(days):
                       for k in range(iterations):
                           avg_vect[k, j]=round(num(p[i],int(avg_vect[k, j-1]),N))
                   amount[i] = np.mean(avg vect, axis=0)[-1]
              amount = amount/N
              plt.plot(p, amount)
In [41]:
          %%time
          limited_avg_num(20, 1, 100, 1000)
         Wall time: 4.15 s
          0.4
          0.3
          0.2
          0.1
          0.0
            0.0000 0.0005 0.0010 0.0015 0.0020 0.0025 0.0030 0.0035 0.0040
In [42]:
          def last():
              N=np.linspace(400,1000,100)
              p_c=1-np.exp(-1/N)
              plt.plot(np.log(p_c), np.log(N))
          %%time
          p_c=[]
          kol=range(400,1000,100)
          for t in kol:
              p_c.append(limited_num(10, 1, 100, t))
          for g in kol:
              g=1/g
          plt.plot(np.log(p_c), np.log(kol))
          last()
         Wall time: 11 s
          6.8
          6.6
          6.4
          6.2
          6.0
                  -6.8
                          -6.6
                                  -6.4
                                          -6.2
                                                   -6.0
```

import matplotlib.pyplot as plt