```python
# Exercise 4.3ef

import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import qr
import sys

title = '4.3e'

a = 1.4
b = 0.3

cut_tail = 100

T = 2*10**5
t_array = np.arange(0,T,1)

x = np.zeros_like(t_array,dtype=float)
y = x.copy()

x[0] = (np.random.uniform()-0.5)
y[0] = (np.random.uniform()-0.5)

for t in range(len(x)-1):
    x[t+1] = y[t]+1-a*x[t]**2
    y[t+1] = b*x[t]

x = x[cut_tail:]
y = y[cut_tail:]

I = np.identity(2)
Q = I.copy()
M = I.copy()

lambda_one = np.zeros_like(x)
lambda_two = lambda_one.copy()

new_T = len(x)
new_t_array = t_array[cut_tail:]-cut_tail

for t in range(len(x)):

    J11 = 2*a*x[t]
    J12 = 1
    J21 = b
    J22 = 0

    J = np.array([[J11,J12],\
                  [J21,J22]])

    M = J
    Q,R = qr(np.matmul(M,Q))

    lambda_one[t] = lambda_one[t-1] + np.log(np.absolute(R[0,0]))/new_T
    lambda_two[t] = lambda_two[t-1] + np.log(np.absolute(R[1,1]))/new_T

lambda_one_converged = lambda_one[-1]
lambda_two_converged = lambda_two[-1]

if lambda_one_converged < lambda_two_converged:
    temp = lambda_one_converged
    lambda_one_converged = lambda_two_converged
    lambda_two_converged = temp

lyapunov_dimension = 1 + lambda_one_converged/np.abs(lambda_two_converged)

print('Lyapunov exponent 1: ', lambda_one_converged)
print('Lyapunov exponent 2: ', lambda_two_converged)
print('Lyapunov dimension: ', lyapunov_dimension)

fig2, ax2 = plt.subplots(figsize=(7,7))
ax2.plot(new_t_array[1:], np.divide(lambda_one[1:],new_t_array[1:]), '-', linewidth=1.5, label=r'$\lambda_1$ (conv. at $\approx$' + '{})'.format(np.round(lambda_one_converged,2)))
ax2.plot(new_t_array[1:], np.divide(lambda_two[1:],new_t_array[1:]), '-', linewidth=1.5, label=r'$\lambda_2$ (conv. at $\approx$' + '{})'.format(np.round(lambda_two_converged,2)))
ax2.set_xscale('log')
```

```python
ax2.set_xlabel('$t$')
ax2.set_ylabel('$\sum \lambda_i$ /t')
ax2.set_box_aspect(1)
ax2.set_title(title)

plt.legend(loc="lower right", prop={'size': 10})
plt.savefig('Dynamical Systems/DS HW4/4.3/'+title+'.png')
plt.show()
```