In [1]:

```
import pandas as pd
import MultiContagion as mc
import igraph
import random
import numpy as np
from matplotlib import pylab as plt
import copy
%matplotlib inline
```

In [2]:

```
pd.read_stata("CPIS_9countries.dta")
```

Out[2]:

	country	countrycode	counterpart	counterpartcode	year	I_A_D_T_T_BP6_USD	I_A_E_T_T_B
0	China, P.R.: Hong Kong	532	China, P.R.: Mainland	924	2005	6.511577e+03	3.479020e+04
1	China, P.R.: Hong Kong	532	Euro Area	163	2005	4.051570e+04	1.531558e+04
2	China, P.R.: Hong Kong	532	Japan	158	2005	7.554724e+03	9.129184e+03

In [3]:

```
pd.read_stata("CDIS_9countries.dta")
```

Out[3]:

	country	countrycode	counterpart	counterpartcode	year	IIWD_BP6_USD	IIWF_BP6_USD	IIW
0	China, P.R.: Hong Kong	532	Euro Area	163	2008	25.731326	0.000000	2.5
1	China, P.R.: Hong Kong	532	ROW	999	2008	0.000000	0.000000	1.9
2	China, P.R.: Hong Kong	532	China, P.R.: Mainland	924	2009	2675.649539	-395.332345	3.1

- I_A_D_TT - is Assets, Debt Securities
- I_A_E_TT - is Assets, Equity Securities
- I_A_T_TT - is Assets, Total Securities
- I_L_D_TT - is Liabilities, Debt Securities
- I_L_E_TT - is Liabilities, Equity Securities
- I_L_T_TT - is Liabilities, Total Securities
- Asset side is reported by most countries, liability side reporting is voluntary.

Mapping CPIS and obtaining basic measurements

In [4]:

```
df_cp = pd.read_stata("CPIS_9countries.dta")
```

In [5]:

```
edges_cp_debt, weight_cp_debt = mc.make_edge_list(df_cp["country"], df_cp["counterp
```

In [6]:

```
G_cp_debt = mc.make_graph_from_edge(edges_cp_debt, weight_cp_debt)
```

In [7]:

```
G_cp_debt.strength(weights=G_cp_debt.es["weight"]);
```

In [8]:

```
G_cp_debt.vs["name"];
```

In [9]:

```
country_name = copy.deepcopy(G_cp_debt.vs["name"])
```

In [10]:

```
max_weight = max(G_cp_debt.es["weight"])
E_width_cp = [0.2 + 5*G_cp_debt.es["weight"][i]/max_weight for i in range(len(G_cp_
```

In [11]:

```
#random.seed(105)
```

In [12]:

```
P_cp_debt = igraph.plot(G_cp_debt, vertex_label = G_cp_debt.vs["name"], bbox = (450
```

In [13]:

```
#P_cp_debt.save("Toy_CPIS_debt.png")
```

In [14]:

```

```

In [15]:

```
#random.seed(105)
edges_cp_equity, weight_cp_equity = mc.make_edge_list(df_cp["country"], df_cp["coun
G_cp_equity = mc.make_graph_from_edge(edges_cp_equity, weight_cp_equity)
max_weight = max(G_cp_equity.es["weight"])
E_width_cp = [0.2 + 5*G_cp_equity.es["weight"][i]/max_weight for i in range(len(G_c
```

In [16]:

```
P_cp_equity = igraph.plot(G_cp_equity, vertex_label = G_cp_equity.vs["name"], bbox
```

In [17]:

```
#P_cp_equity.save("Toy_CPIS_equity.png")
```

In [41]:

```
str_cp_debt = G_cp_debt.strength(weights=G_cp_debt.es["weight"])
in_str_cp_debt = G_cp_debt.strength(weights=G_cp_debt.es["weight"], mode = "IN")
out_str_cp_debt = G_cp_debt.strength(weights=G_cp_debt.es["weight"], mode = "OUT")
```

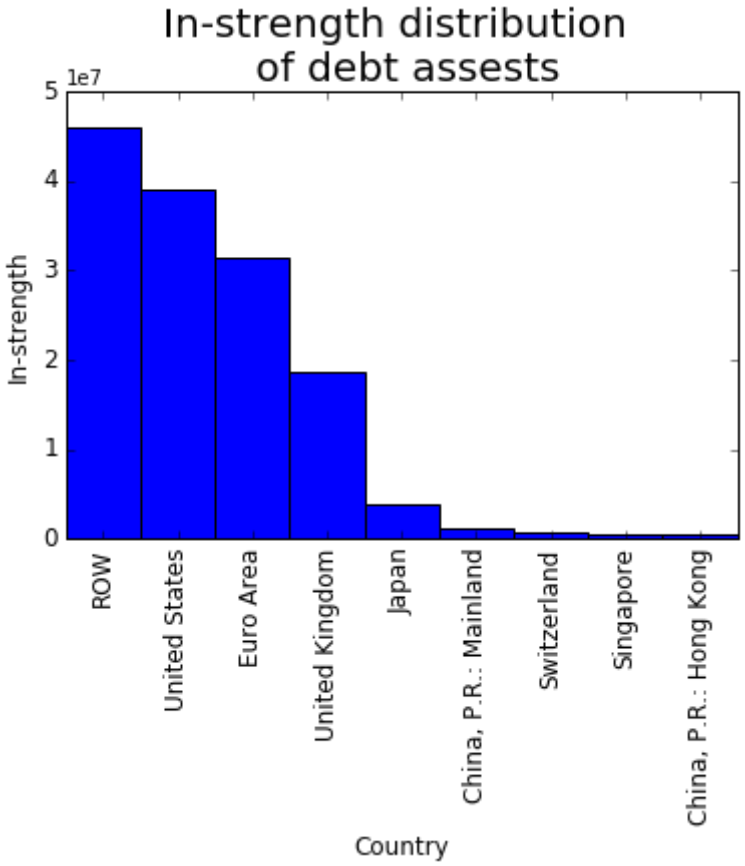
In [42]:

```
names = list(reversed([x for (y, x) in sorted(zip(in_str_cp_debt, country_name))]))
```

In [43]:

```
plt.figure(figsize=(10,10))
fig, ax = plt.subplots()
ax1 = plt.gca()
names = list(reversed([x for (y, x) in sorted(zip(in_str_cp_debt, country_name))]))
in_str_cp_debt_ordered = list(reversed(sorted(in_str_cp_debt)))
ax1.bar(np.arange(9), in_str_cp_debt_ordered, 1)
plt.title("In-strength distribution \n of debt assests", fontsize = 20)
plt.xticks(0.5 + np.arange(9), names, rotation='vertical', fontsize = 12)
plt.yticks(fontsize = 12)
plt.xlabel("Country", fontsize = 12)
plt.ylabel("In-strength", fontsize = 12)
#plt.savefig("instr_cp_debt.png",bbox_inches='tight')
plt.show()
```

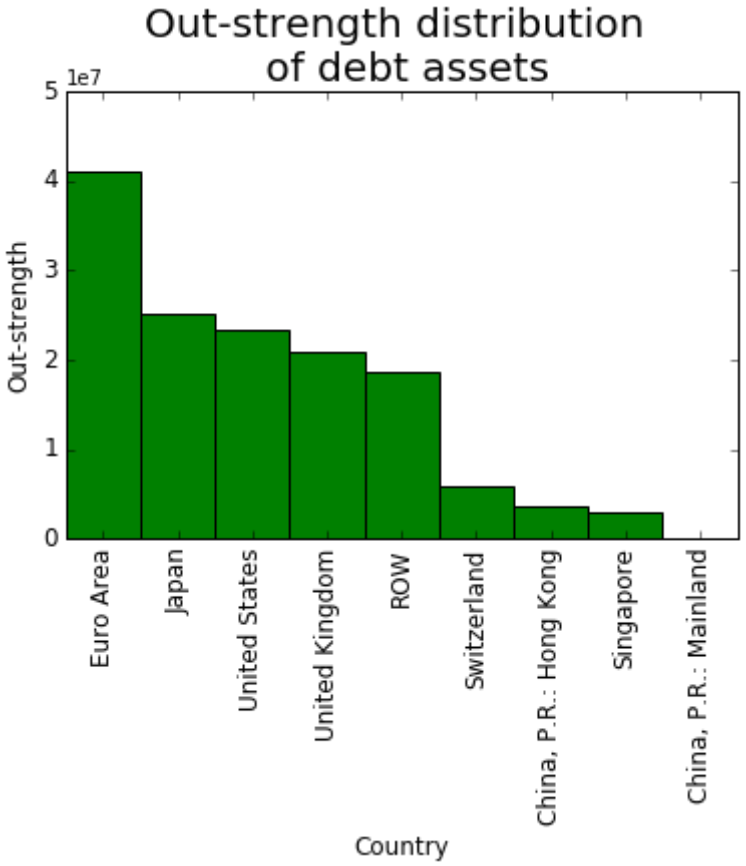
<matplotlib.figure.Figure at 0x7fba3eedee10>



In [56]:

```
plt.figure(figsize=(10,10))
fig, ax = plt.subplots()
ax1 = plt.gca()
names = list(reversed([x for (y, x) in sorted(zip(out_str_cp_debt, country_name))])
out_str_cp_debt_ordered = list(reversed(sorted(out_str_cp_debt)))
ax1.bar(np.arange(9), out_str_cp_debt_ordered, 1, color = "green")
plt.title("Out-strength distribution \n of debt assets", fontsize = 20)
plt.xticks(0.5 + np.arange(9),names, rotation='vertical', fontsize = 12)
plt.yticks(fontsize = 12)
plt.ylim([0, 5e7])
plt.xlabel("Country", fontsize = 12)
plt.ylabel("Out-strength", fontsize = 12)
#plt.savefig("outstr_cp_debt.png",bbox_inches='tight')
plt.show()
```

```
<matplotlib.figure.Figure at 0x7fc98c29d290>
```



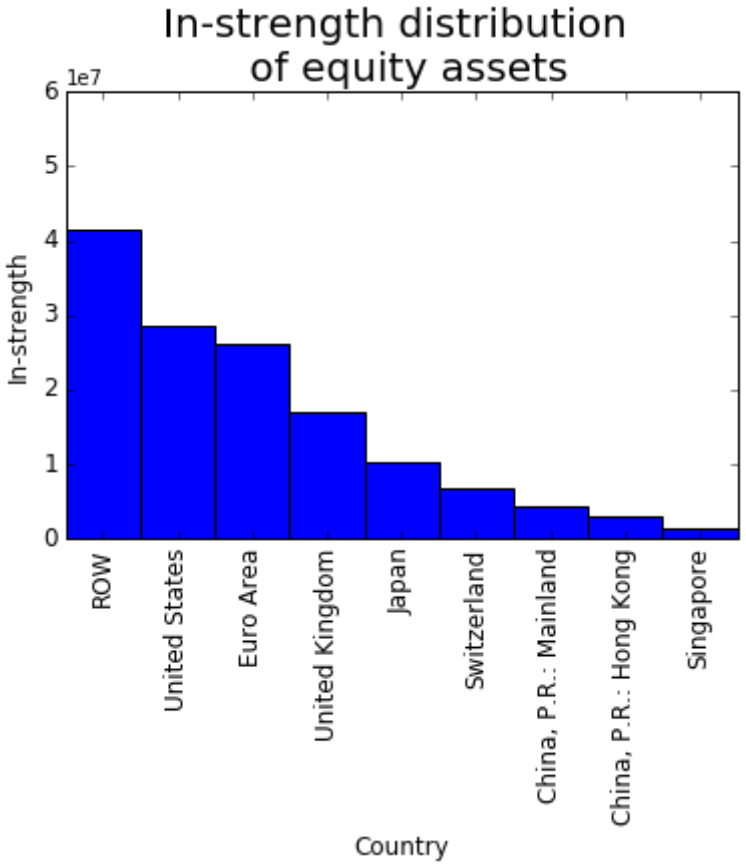
In [8]:

```
str_cp_equity = G_cp_debt.strength(weights=G_cp_equity.es["weight"])
in_str_cp_equity = G_cp_debt.strength(weights=G_cp_equity.es["weight"], mode = "IN"
out_str_cp_equity = G_cp_debt.strength(weights=G_cp_equity.es["weight"], mode = "OU
```

In [57]:

```
plt.figure(figsize=(10,10))
fig, ax = plt.subplots()
ax1 = plt.gca()
names = list(reversed([x for (y, x) in sorted(zip(in_str_cp_equity, country_name))]
in_str_cp_equity_ordered = list(reversed(sorted(in_str_cp_equity)))
ax1.bar(np.arange(9), in_str_cp_equity_ordered, 1)
plt.title("In-strength distribution \n of equity assets", fontsize = 20)
plt.xticks(0.5 + np.arange(9), names, rotation='vertical', fontsize = 12)
plt.yticks(fontsize = 12)
plt.ylim([0, 6e7])
plt.xlabel("Country", fontsize = 12)
plt.ylabel("In-strength", fontsize = 12)
#plt.savefig("instr_cp_equity.png",bbox_inches='tight')
plt.show()
```

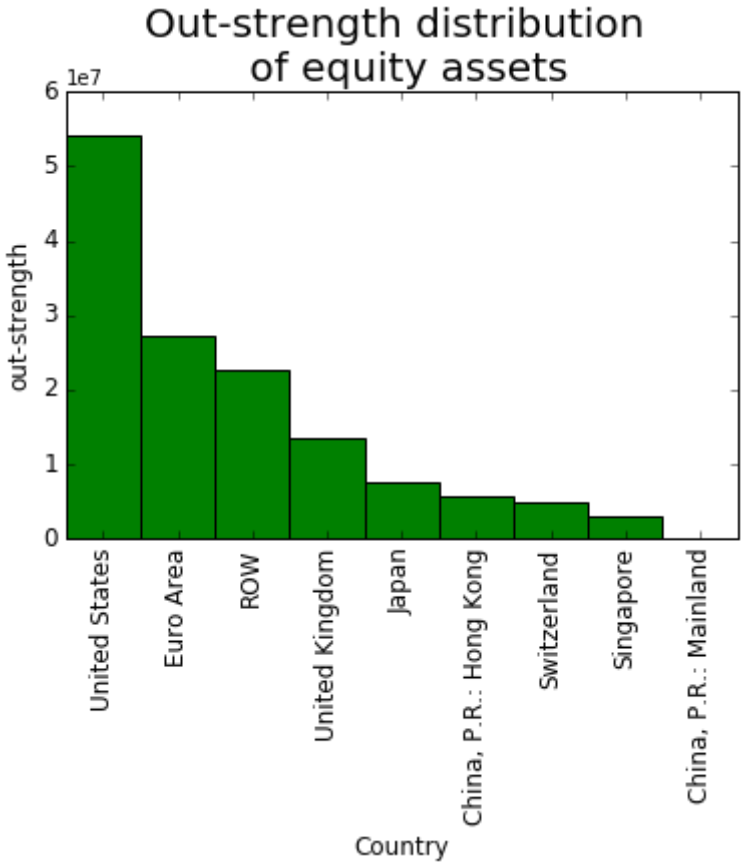
<matplotlib.figure.Figure at 0x7fc98beb4a90>



In [12]:

```
plt.figure(figsize=(10,10))
fig, ax = plt.subplots()
ax1 = plt.gca()
names = list(reversed([x for (y, x) in sorted(zip(out_str_cp_equity, country_name))
out_str_cp_equity_ordered = list(reversed(sorted(out_str_cp_equity)))
ax1.bar(np.arange(9), out_str_cp_equity_ordered, 1, color = "green")
plt.title("Out-strength distribution \n of equity assets", fontsize = 20)
plt.xticks(0.5 + np.arange(9), names, rotation='vertical', fontsize = 12)
plt.yticks(fontsize = 12)
plt.ylim([0, 6e7])
plt.xlabel("Country", fontsize = 12)
plt.ylabel("out-strength", fontsize = 12)
#plt.savefig("outstr_cp_equity.png",bbox_inches='tight')
plt.show()
```

```
<matplotlib.figure.Figure at 0x7f26ad209ad0>
```



Contagion on CPIS

In [20]:

```
mc.SI_contagion_time(G_cp_debt, lam = 0.5, chosen_one=2)
```

Out[20]:

```
(7.6699999999999999, 2.6535071132371213)
```

In [21]:

```
lam_list = [0.25, 0.5, 0.75]
T_iter = 100
n_countries = 9
```

In [22]:

```
def make_several_contagion_list(G, contagion_time = mc.SI_contagion_time, parameter
    cont_time_mean_list = []
    cont_time_std_list = []
    for i in range(len(parameters)):
        cont_time_mean_list.append([])
        cont_time_std_list.append([])
        lam = parameters[i]
        for node in range(n_countries):
            m, s = contagion_time(G_cp_debt, lam = lam, chosen_one= node, iteration
            cont_time_mean_list[i].append(m)
            cont_time_std_list[i].append(s)

    return cont_time_mean_list, cont_time_std_list
```

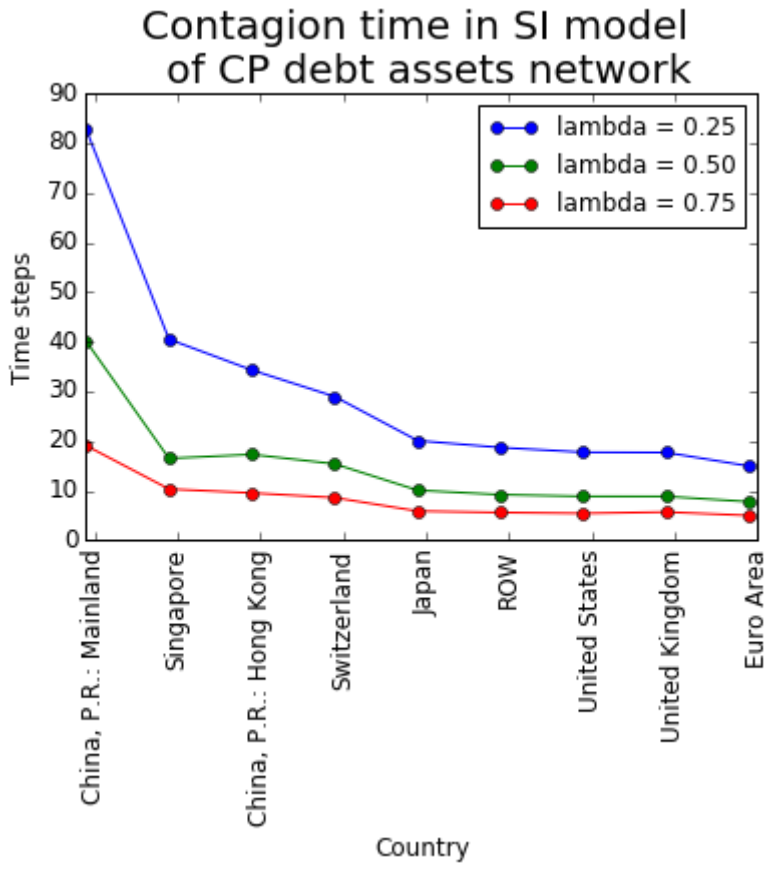
In [37]:

```
SI_cont_time_mean_list, SI_cont_time_std_list  = make_several_contagion_list(G_cp_d
```

In [38]:

```
plt.figure(figsize=(10,10))
fig, ax = plt.subplots()
ax1 = plt.gca()
names = list(reversed([x for (y, x) in sorted(zip(SI_cont_time_mean_list[0], countr
SI_cont_time_mean_list_ordered2 = list(reversed([x for (y, x) in sorted(zip(SI_cont
SI_cont_time_mean_list_ordered3 = list(reversed([x for (y, x) in sorted(zip(SI_cont
SI_cont_time_mean_list_ordered1 = list(reversed(sorted(SI_cont_time_mean_list[0])))
ax1.plot(SI_cont_time_mean_list_ordered1, "o-", label = "lambda = 0.25")
ax1.plot(SI_cont_time_mean_list_ordered2, "o-", label = "lambda = 0.50")
ax1.plot(SI_cont_time_mean_list_ordered3, "o-", label = "lambda = 0.75")
plt.title("Contagion time in SI model \n of CP debt assets network", fontsize = 20)
plt.xticks(0.1 + np.arange(9), names, rotation='vertical', fontsize = 12)
plt.yticks(fontsize = 12)
#plt.ylim([0, 6e7])
plt.xlabel("Country", fontsize = 12)
plt.ylabel("Time steps", fontsize = 12)
plt.legend()
#plt.savefig("SIcont-time_cp_debt.png",bbox_inches='tight')
plt.show()
```

```
<matplotlib.figure.Figure at 0x7fba3eb93990>
```



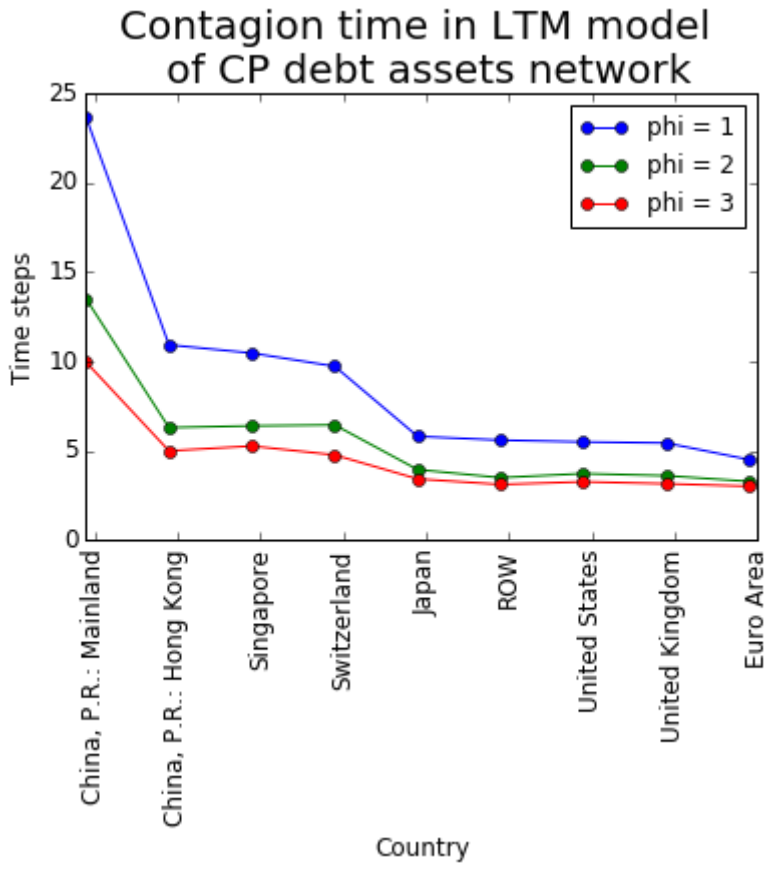
In [39]:

```
LTM_cont_time_mean_list, LTM_cont_time_std_list  = make_several_contagion_list(G_cp
```

In [40]:

```
plt.figure(figsize=(10,10))
fig, ax = plt.subplots()
ax1 = plt.gca()
names = list(reversed([x for (y, x) in sorted(zip(LTM_cont_time_mean_list[0], count
LTM_cont_time_mean_list_ordered2 = list(reversed([x for (y, x) in sorted(zip(LTM_co
LTM_cont_time_mean_list_ordered3 = list(reversed([x for (y, x) in sorted(zip(LTM_co
LTM_cont_time_mean_list_ordered1 = list(reversed(sorted(LTM_cont_time_mean_list[0])
ax1.plot(LTM_cont_time_mean_list_ordered1, "o-", label = "phi = 1")
ax1.plot(LTM_cont_time_mean_list_ordered2, "o-", label = "phi = 2")
ax1.plot(LTM_cont_time_mean_list_ordered3, "o-", label = "phi = 3")
plt.title("Contagion time in LTM model \n of CP debt assets network", fontsize = 20
plt.xticks(0.1 + np.arange(9), names, rotation='vertical', fontsize = 12)
plt.yticks(fontsize = 12)
#plt.ylim([0, 6e7])
plt.xlabel("Country", fontsize = 12)
plt.ylabel("Time steps", fontsize = 12)
plt.legend()
#plt.savefig("LTMcont-time_cp_debt.png",bbox_inches='tight')
plt.show()
```

```
<matplotlib.figure.Figure at 0x7fba41e3b5d0>
```



For Equity data

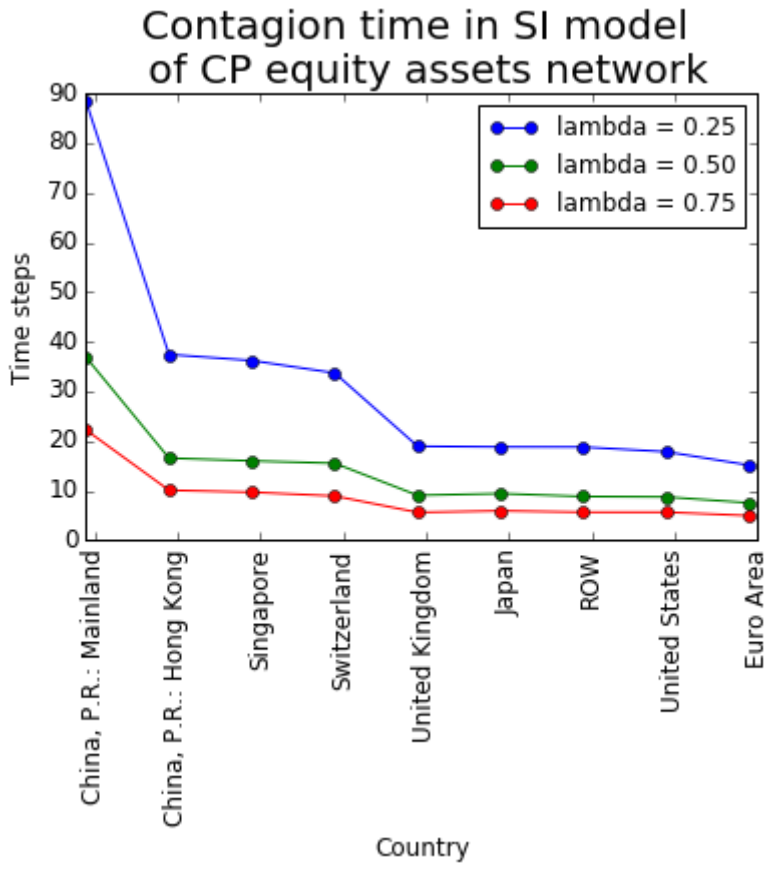
In [ ]:

```
SI_cont_time_mean_list, SI_cont_time_std_list  = make_several_contagion_list(G_cp_e
```

In [36]:

```
plt.figure(figsize=(10,10))
fig, ax = plt.subplots()
ax1 = plt.gca()
names = list(reversed([x for (y, x) in sorted(zip(SI_cont_time_mean_list[0], countr
SI_cont_time_mean_list_ordered2 = list(reversed([x for (y, x) in sorted(zip(SI_cont
SI_cont_time_mean_list_ordered3 = list(reversed([x for (y, x) in sorted(zip(SI_cont
SI_cont_time_mean_list_ordered1 = list(reversed(sorted(SI_cont_time_mean_list[0])))
ax1.plot(SI_cont_time_mean_list_ordered1, "o-", label = "lambda = 0.25")
ax1.plot(SI_cont_time_mean_list_ordered2, "o-", label = "lambda = 0.50")
ax1.plot(SI_cont_time_mean_list_ordered3, "o-", label = "lambda = 0.75")
plt.title("Contagion time in SI model \n of CP equity assets network", fontsize = 2
plt.xticks(0.1 + np.arange(9), names, rotation='vertical', fontsize = 12)
plt.yticks(fontsize = 12)
#plt.ylim([0, 6e7])
plt.xlabel("Country", fontsize = 12)
plt.ylabel("Time steps", fontsize = 12)
plt.legend()
plt.savefig("SIcont-time_cp_equity.png",bbox_inches='tight')
plt.show()
```

```
<matplotlib.figure.Figure at 0x7fba3f05f510>
```



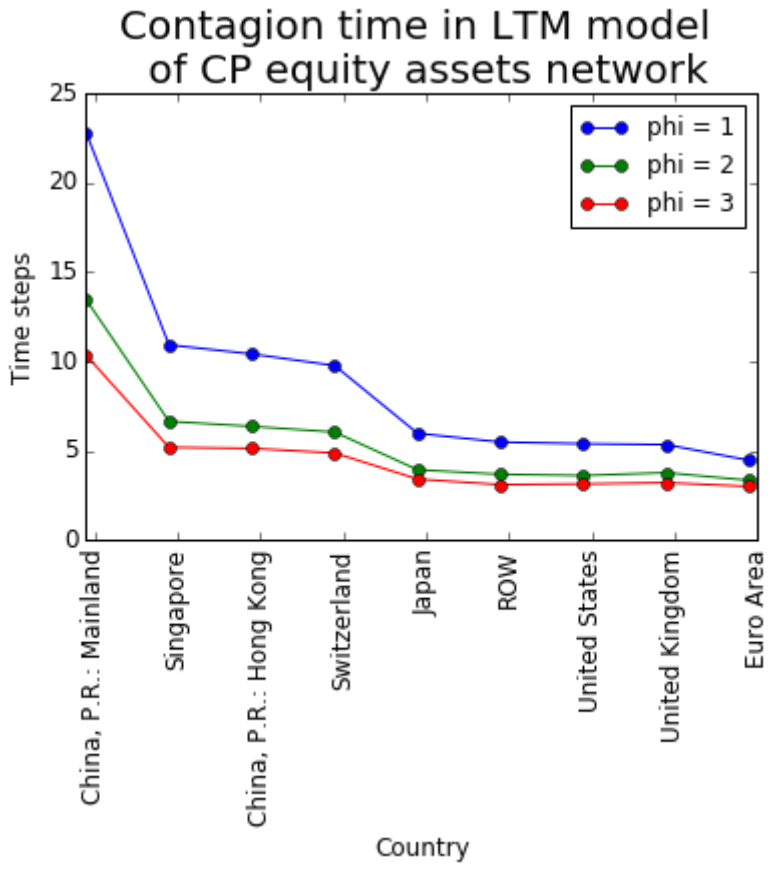
In [33]:

```
LTM_cont_time_mean_list, LTM_cont_time_std_list  = make_several_contagion_list(G_cp
```

In [35]:

```
plt.figure(figsize=(10,10))
fig, ax = plt.subplots()
ax1 = plt.gca()
names = list(reversed([x for (y, x) in sorted(zip(LTM_cont_time_mean_list[0], count
LTM_cont_time_mean_list_ordered2 = list(reversed([x for (y, x) in sorted(zip(LTM_co
LTM_cont_time_mean_list_ordered3 = list(reversed([x for (y, x) in sorted(zip(LTM_co
LTM_cont_time_mean_list_ordered1 = list(reversed(sorted(LTM_cont_time_mean_list[0])
ax1.plot(LTM_cont_time_mean_list_ordered1, "o-", label = "phi = 1")
ax1.plot(LTM_cont_time_mean_list_ordered2, "o-", label = "phi = 2")
ax1.plot(LTM_cont_time_mean_list_ordered3, "o-", label = "phi = 3")
plt.title("Contagion time in LTM model \n of CP equity assets network", fontsize =
plt.xticks(0.1 + np.arange(9), names, rotation='vertical', fontsize = 12)
plt.yticks(fontsize = 12)
#plt.ylim([0, 6e7])
plt.xlabel("Country", fontsize = 12)
plt.ylabel("Time steps", fontsize = 12)
plt.legend()
plt.savefig("LTMcont-time_cp_equity.png",bbox_inches='tight')
plt.show()
```

<matplotlib.figure.Figure at 0x7fba3eb54850>



In [ ]: