A Data-driven approach to find links between communities using the Graph's snapshot subgraph API

In [55]:
```
# import libraries

import requests
import json
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
# Production hub API found on https://docs.snapshot.org/guides/graphql-api

snapshot_url = 'https://hub.snapshot.org/graphql'
```

In [3]:
```
# Function taking as input
# 1) lim: first: representing the first indexed proposal
# 2) off: skip: represennting the skipped indexed proposal
# Returns the query to be run

def query_(lim,off):
    query = """query Votes2 {
      votes(
        first: """ + str(lim) +""",
        skip: """ + str(off) +""",
      )
      {
        voter,
        space {
          name
        }
      }
    }"""
    return query
```

In [36]:
```
# for loop querying 1000 proposals per loop

df = pd.DataFrame()

for x in range(1000, 6000, 1000):

    try:
        r = requests.post(snapshot_url, json={'query': query_(x,x-1000)})
        json_data = json.loads(r.text)

        space_lst = []
        member_lst = []

        index = 0

        for i in range(len(json_data['data']['votes'])):
            space_lst.append(json_data['data']['votes'][i]['space']['name'])
            member_lst.append(json_data['data']['votes'][i]['voter'])

        df = df.append(pd.DataFrame([space_lst, member_lst]).T)
    except:
        continue

df.columns = ['DAO', 'Contributors']
df = df.drop_duplicates()
```

In [37]:
```
# dataframe containing contributors addresses to their DAO

df
```

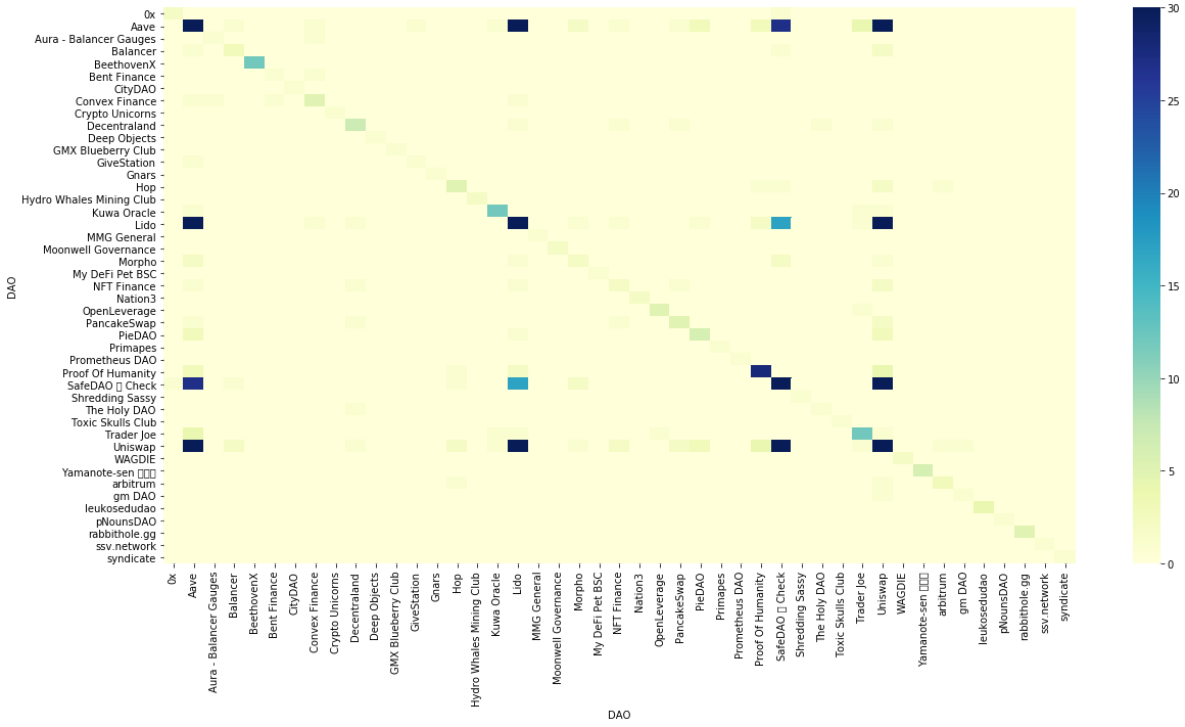
Out[37]:

	DAO	Contributors
0	Hydro Whales Mining Club	0x00907204C6EbD3706A480A2dA0e4A06ea18E4111
1	Prometheus DAO	0xa04A4c881516c49B0A16523cC8D4328688E21c9a
2	Uniswap	0x7edb74a70ADcaaC6b739b5610Ea311D44c628015
3	Kuwa Oracle	0xdA3ac2621E9Cd980A7C05aCb7d901CfFA98fBE36
4	Moonwell Governance	0x533be0e909309aDC0bc9FBc95692fa5bc42eB8d6
...	...	...
994	Aave	0x5DeE6eD1506D08809A541D1945121e7626EdC748
996	Nation3	0x3B8eF2Bc9c975EAd87FE2708e38aCCa7BE98df7f
997	Aave	0x0DbaF867815BA6Eb831A433a28c401575fC904b9
998	Uniswap	0x8C72B6ABdA7b68808851de83e3e59FD47B8c4609
999	Uniswap	0x560Bc68D73ec2d0C25eB9a655eF915d9979F5E42

688 rows × 2 columns

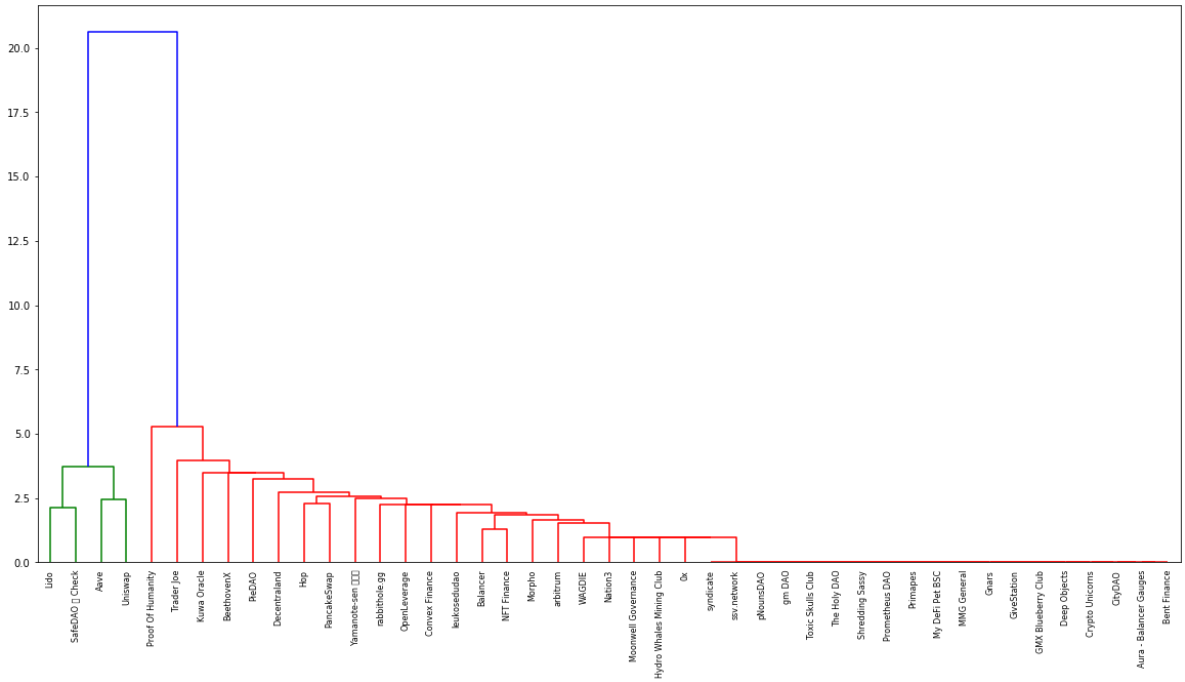
In [39]:

```
# Create a cross-tabulation table
cross_tab = pd.crosstab(index=df['DAO'], columns=df['Contributors'])

# Multiply the cross-tabulation table by its transpose to get the 2D matrix
matrix = cross_tab.dot(cross_tab.T)

# plot
plt.figure(figsize=(20,10))

vmin = 0
vmax = 30

# Create a heatmap using seaborn
sns.heatmap(matrix, cmap="YlGnBu", vmin=vmin, vmax=vmax)

# Show the plot
plt.show()
```



In [102]:
```
import vg;
from scipy.cluster.hierarchy import dendrogram, linkage;

z = linkage(np.log(matrix[np.log(matrix)>0]).fillna(0), method='ward', metric
='euclidean');

plt.figure(figsize=(20,10));

# Plot the dendrogram;
dendrogram(z, labels=matrix.index);
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:4: RuntimeWar
ning: divide by zero encountered in log
  after removing the cwd from sys.path.
```



Litterature of future work

https://www.pnas.org/doi/10.1073/pnas.122653799 (https://www.pnas.org/doi/10.1073/pnas.122653799)