

```
In [48]: from random import randint
import pandas as pd
from typing import List
import matplotlib.pyplot as plt
from networkx import density
import networkx as nx
from networkx import number_connected_components
from networkx import average_degree_connectivity
import networkx.algorithms.community as nxcom
from networkx.algorithms.community.centralities import girvan_newman
from collections import Counter
import collections
import math
import community
import numpy as np
import csv
import community as louvain_community
import numpy as np
from math import log
```

```
In [49]: nodes = pd.read_csv('nodes.csv')
edges = pd.read_csv('edges.csv')
cleanedges = edges[['Source', 'Target']]
cleannodes = nodes[['Id', 'Label']]
```

```
In [50]: cleanedges
```

```
Out[50]:
```

	Source	Target
0	606935	1420000
1	606935	14785048
2	606935	21759323
3	606935	27835117
4	606935	48038134
...
839	255783283	343057187
840	255783283	546700624
841	297264347	546700624
842	298914898	540038266
843	343057187	546700624

844 rows × 2 columns

In [51]: cleannodes

Out[51]:

	Id	Label
0	606935	Anna Kovalova
1	1420000	Maria Krivocheina
2	1687066	Guerman Guermanovitch
3	1895348	Daniil Skorinkine
4	1922205	Aleksandra Kachperskaia
...
129	393042942	Danila Tsekhovskoi
130	467822313	Evgueni Glazounov
131	540038266	Irina Loguinova
132	546700624	Elia Roditcheva
133	591276730	Daniil Voronov

134 rows × 2 columns

```
In [52]: nodesdict = {row[1]['Id'] : row[1]['Label'] for row in cleannodes.iterrows()}
clean_labels = cleanedges.applymap(lambda x: nodesdict[x])

for row in cleanedges.itertuples():
    clean_labels.loc[row[0], 'Source'] = nodesdict[row[1]]
    clean_labels.loc[row[0], 'Target'] = nodesdict[row[2]]
```

```
In [53]: clean_labels
```

```
Out[53]:
```

	Source	Target
0	Anna Kovalova	Maria Krivocheina
1	Anna Kovalova	Natalia Dolgoroukova
2	Anna Kovalova	Sacha Anisimov
3	Anna Kovalova	Elizaveta Senatorova
4	Anna Kovalova	Mitia Krylov
...
839	Ilona Moskovskaïa	Kovaliov Alexander
840	Ilona Moskovskaïa	Elia Roditcheva
841	Dacha Lepakhina	Elia Roditcheva
842	Jenia Zakovorotnaïa	Irina Loguinova
843	Kovaliov Alexander	Elia Roditcheva

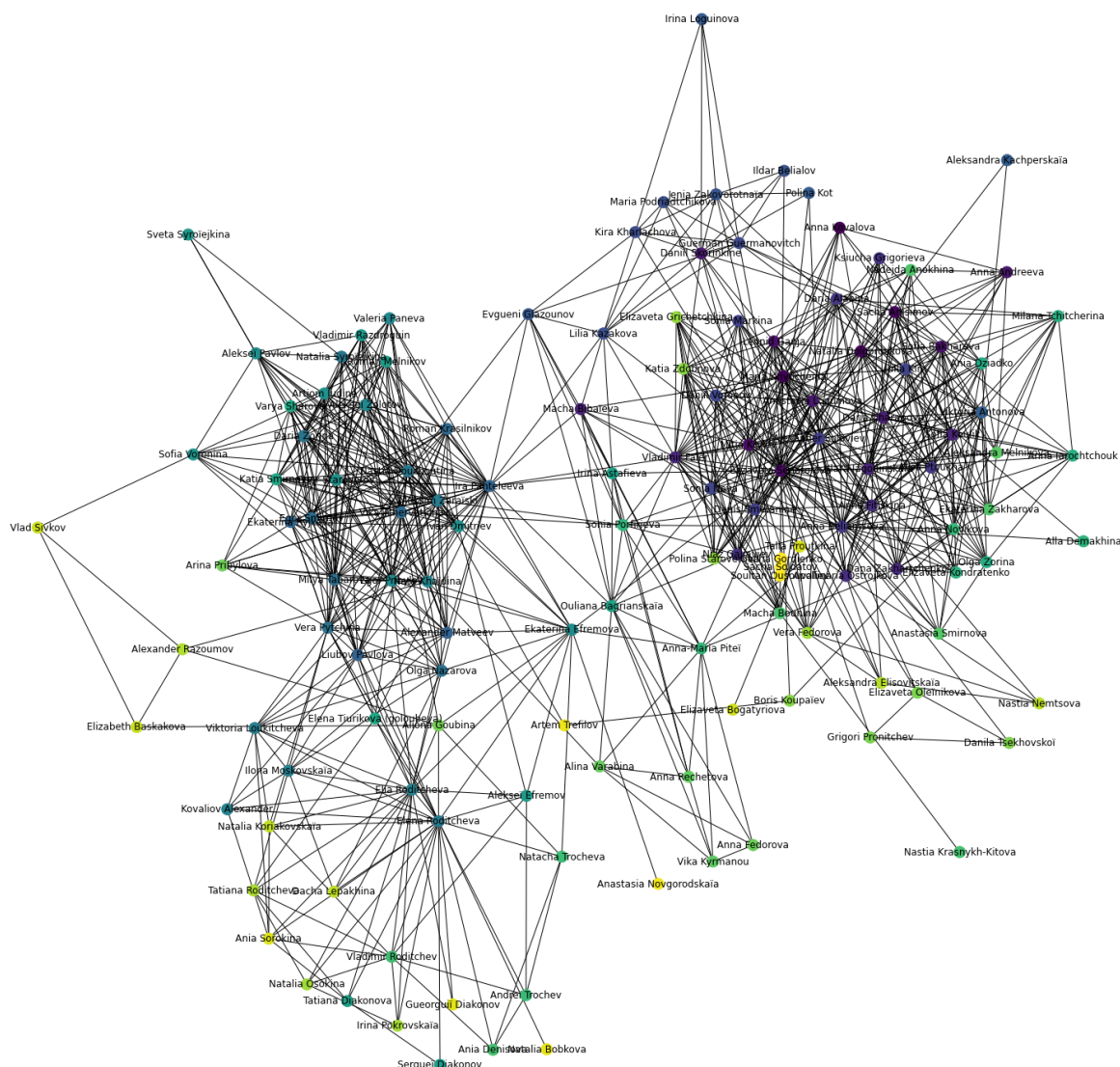
844 rows × 2 columns

```
In [54]: graph = nx.from_pandas_edgelist(clean_labels, 'Source', 'Target')
```

```
In [55]: #добавляем одиночные узлы
newlist = []
for value in dict.values(nodesdict):
    newlist.append(value)

for label in newlist:
    if label not in graph.nodes:
        graph.add_node(label)
```

```
In [56]: nx.kamada_kawai_layout(graph)
pos = nx.kamada_kawai_layout(graph)
plt.figure(figsize=(20, 20))
nx.draw(graph, pos, with_labels=True, node_color=range(graph.number_of_n
odes()),node_size=200)
```



На первый взгляд два плотных подсообщества - школа в родном городе и универ в Москве, многие одноклассники поступили в Вышку, поэтому связи есть между подсообществами в том числе

```
In [57]: print('Количество вершин: {}'.format(graph.number_of_nodes()))
print('Количество рёбер: {}'.format(graph.number_of_edges()))
print('Количество компонент:', nx.number_connected_components(graph))
print('Плотность:', nx.density(graph))
print('Средняя степень вершины: {}'.format(round(graph.number_of_edges() / float(graph.number_of_nodes()), 2)))
```

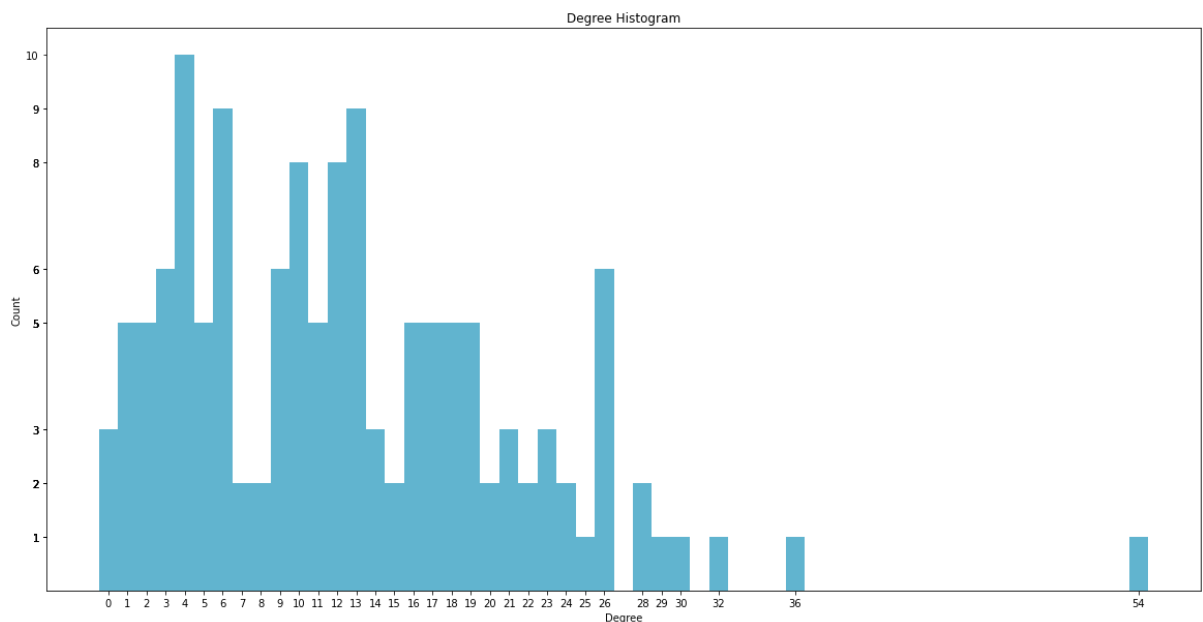
Количество вершин: 134
 Количество рёбер: 844
 Количество компонент: 5
 Плотность: 0.09471439793513635
 Средняя степень вершины: 6.3

5 компонент из-за 3 человек из прошлого, которые добавились в друзья + 1 компонента из двух соседей в новой общаге

```
In [58]: degree_sequence = sorted([degree for node, degree in graph.degree()], reverse=True) # degree sequence
degree_count = Counter(degree_sequence)
degree, count = zip(*degree_count.items())

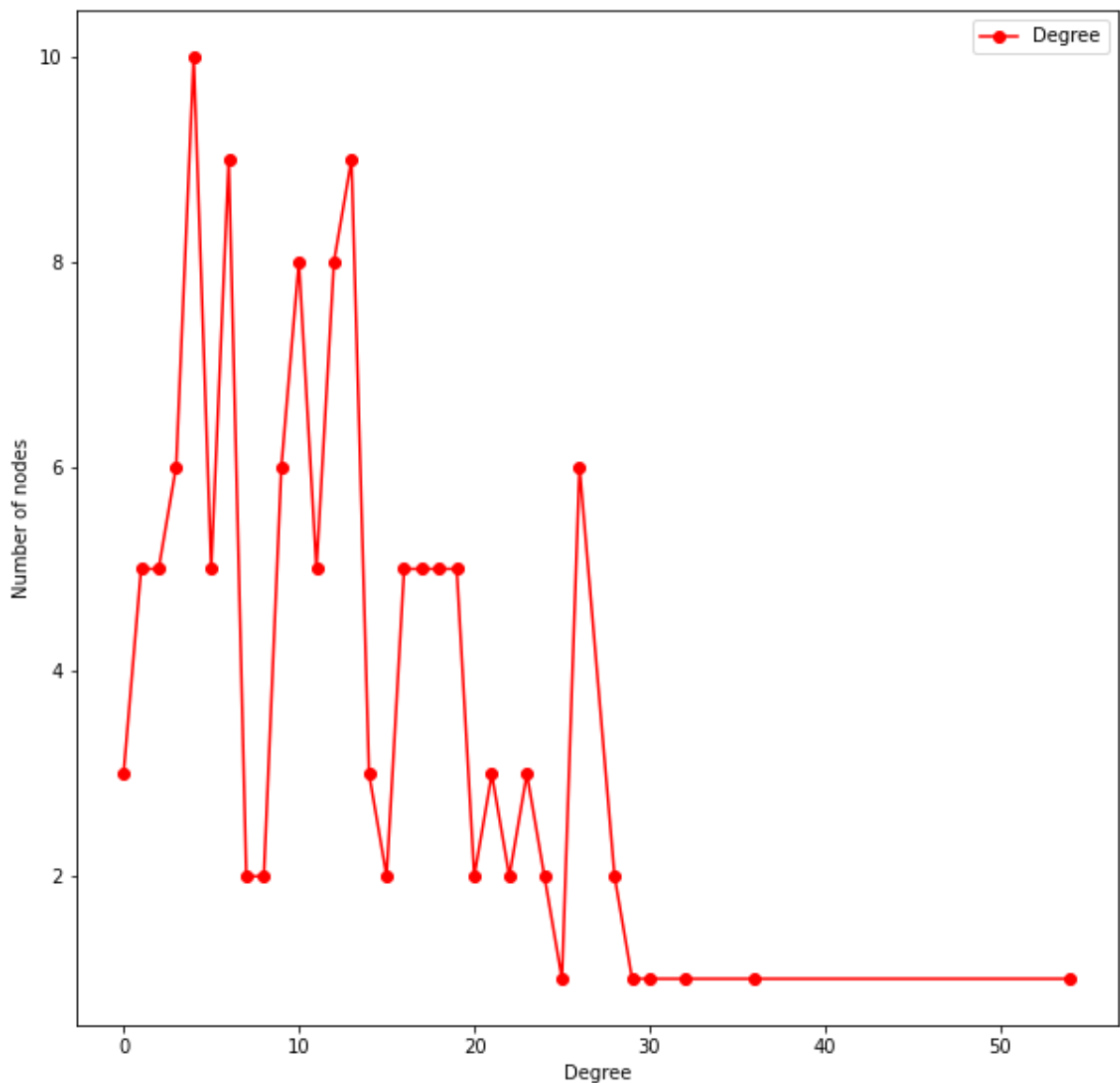
fig, ax = plt.subplots(figsize=(20,10))
plt.bar(degree, count, width=1, color="#61B4CF")

plt.title("Degree Histogram")
plt.ylabel("Count")
plt.xlabel("Degree")
ax.set_xticks(degree)
ax.set_yticks(count)
plt.show()
```



В основном больше 30 связей имеют всего несколько человек из универа, либо близкие друзья, либо преподаватели. В остальном все более-менее равномерно. Я либо плотно общаюсь в компанию из 4 человек, либо около 10 (но это уже университетские и школьные компании)

```
In [59]: degree = dict(graph.degree())
degree_values = sorted(set(degree.values()))
hist = [list(degree.values()).count(x) for x in degree_values]
plt.figure(figsize=(10, 10))
plt.plot(degree_values, hist, 'ro-')
plt.legend(['Degree'])
plt.xlabel('Degree')
plt.ylabel('Number of nodes')
plt.show()
```



```
In [60]: degree = nx.degree centrality(graph)
betweenness = nx.betweenness centrality(graph)
closeness = nx.closeness centrality(graph)
df = pd.DataFrame([degree, closeness, betweenness]), index=['degree', 'closeness', 'betweenness']).T
df
```

Out[60]:

	degree	closeness	betweenness
Anna Kovalova	0.067669	0.347989	0.000770
Maria Krivocheina	0.195489	0.392318	0.006850
Natalia Dolgoroukova	0.180451	0.386169	0.018692
Sacha Anisimov	0.142857	0.358105	0.005419
Elizaveta Senatorova	0.406015	0.506946	0.320060
...
Artem Trefilov	0.015038	0.324179	0.003210
Anastasia Novgorodskaja	0.007519	0.324179	0.000000
Soultan Ousouvaliev	0.000000	0.000000	0.000000
Sacha Soldatov	0.000000	0.000000	0.000000
Gocha Gordienko	0.000000	0.000000	0.000000

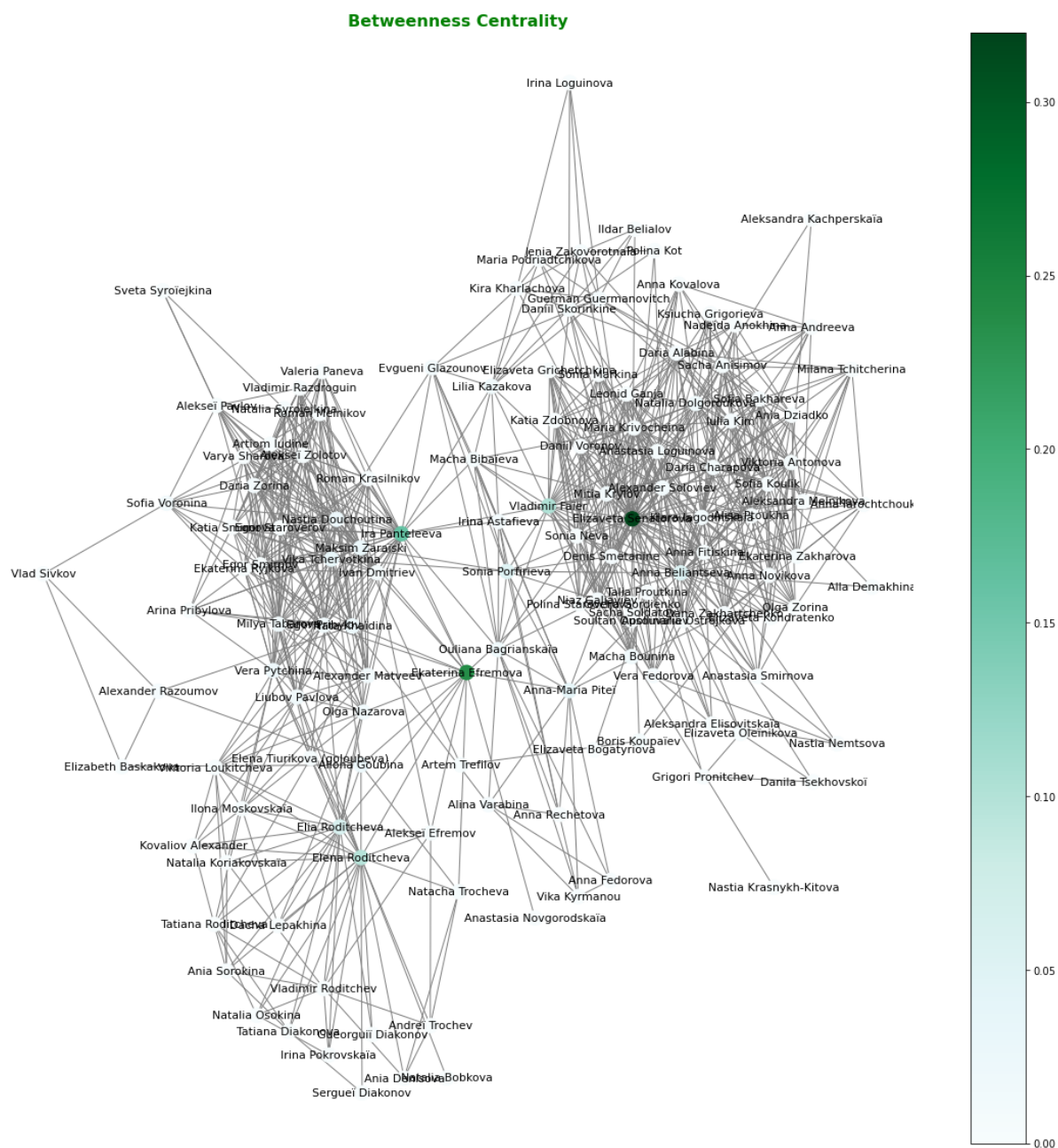
134 rows × 3 columns

```
In [61]: def centr(G, pos, measures, name):

    fig = plt.figure(figsize=(20,20))
    nodes = nx.draw_networkx_nodes(G, pos, node_size=200, cmap=plt.cm.Bu
Gn,
                                node_color=list(measures.values()),
                                nodelist=measures.keys())
    labels = nx.draw_networkx_labels(G, pos, font_color='black', font_size='11')
    edges = nx.draw_networkx_edges(G, pos, edge_color = 'grey')

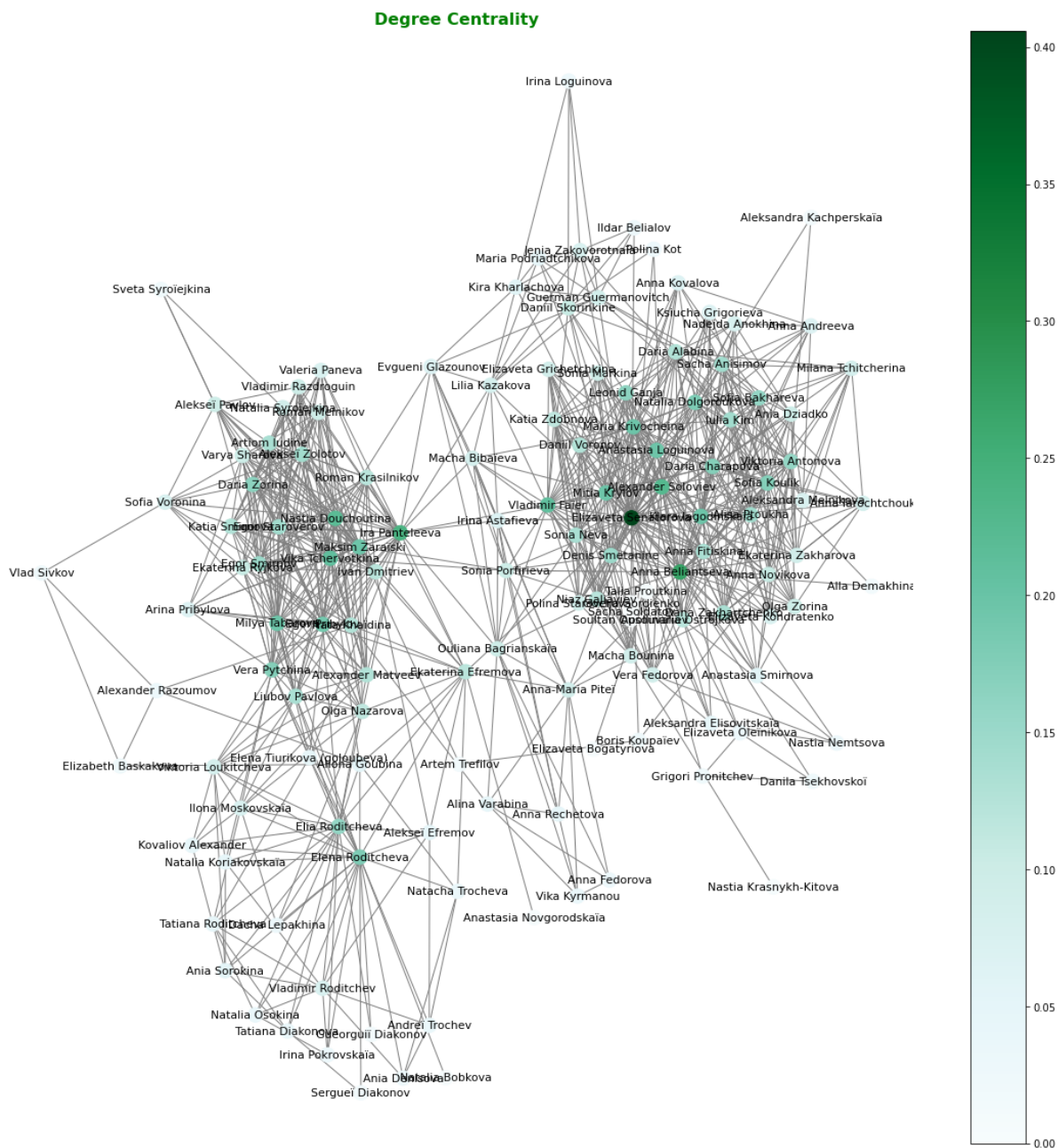
    plt.title(name, color='green', size = '16', weight = 'bold')
    plt.colorbar(nodes, orientation = 'vertical')
    plt.axis('off')
    fig.set_facecolor('white')
    plt.show()
```

```
In [62]: pos = nx.kamada_kawai_layout(graph)
centr(graph, pos, nx.betweenness centrality(graph), 'Betweenness Centrali
ty')
```



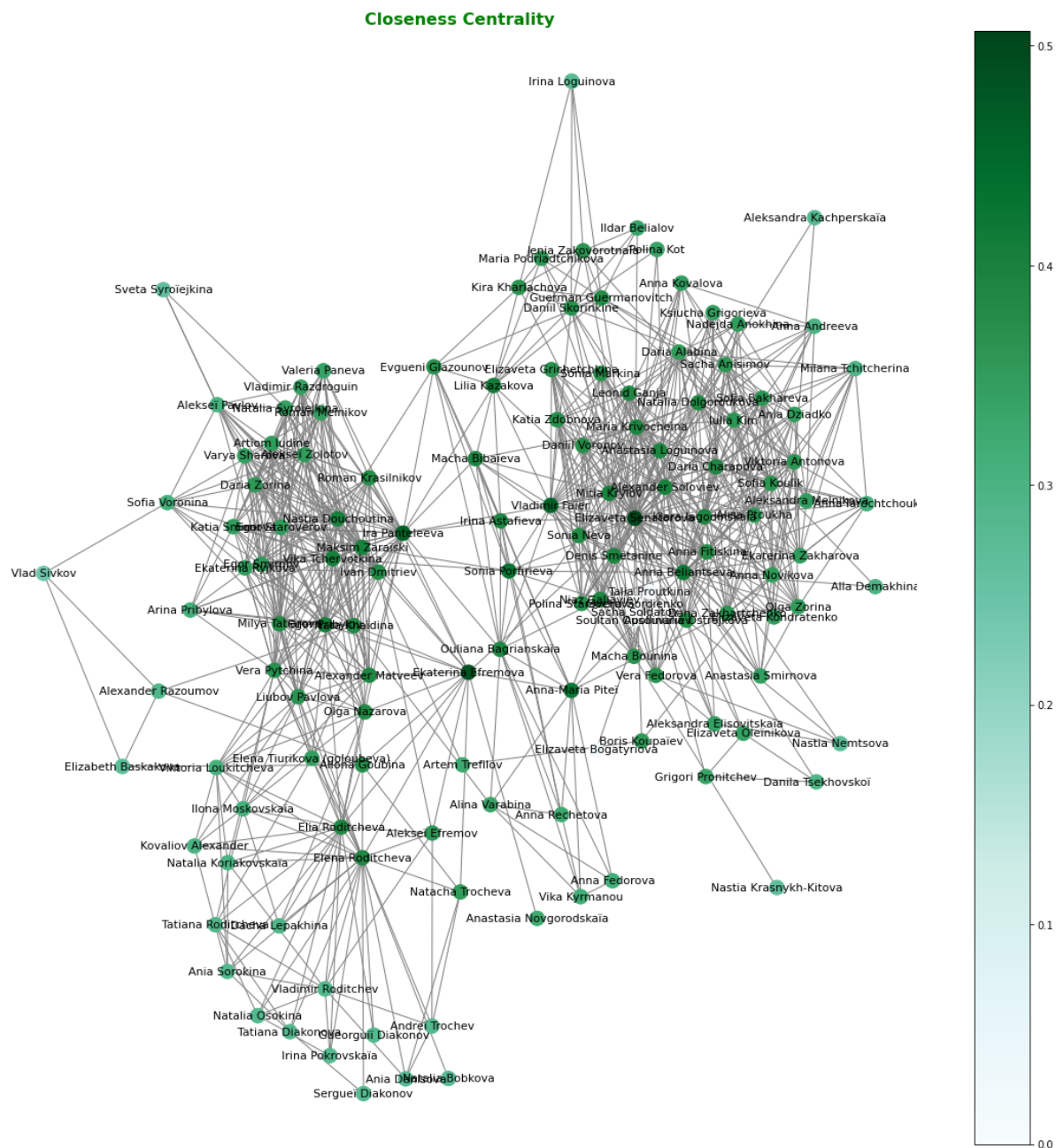
Самая большая центральность здесь (и на всех остальных графах - у Лизы), из общего у нас филфак и магистратура + она знакома с моими друзьями по общеаг и из моего города. Вторая - моя лучшая подруга из моего города, моей школы, которая к тому же знает и моих московских друзей, третья - моя одноклассница, которая поступила в Вышку на лингвистику.


```
In [63]: centr(graph, pos, nx.degree_centrality(graph), 'Degree Centrality')
```



Здесь опять же Лиза и моя подруга с филфака, обе очень общительные и активные, в отличие от меня + мы все из одной компании

```
In [64]: centr(graph, pos, nx.closeness centrality(graph), 'Closeness Centrality')
```



Здесь различия помягче, все более- менее плотно

```
In [65]: cm = nxcom.girvan_newman(graph)
communities = next(cm)
print(len(communities))
```

6

Это похоже на правду. Около 3 сообществ в Москве: бакалавриат, магистратура, общага, и столько же в Вологде: семья, школа и еще одна тусовка

```
In [66]: partition = louvain_community.best_partition(graph)
modularity = louvain_community.modularity(partition, graph)

print('Количество сообществ:', max(Counter(partition).values()))
print('Модулярность:', modularity)

fig = plt.figure(figsize = (10,10))
values = [partition.get(node) for node in graph.nodes()]
nx.draw_spring(graph, cmap=plt.cm.BuGn, node_color = values, node_size=10, with_labels=True, font_color = 'black', edge_color = 'dimgray')
plt.title('Modularity', size = '16', weight = 'bold', color = 'green')
fig.set_facecolor('white')
plt.show()
```

Количество сообществ: 10

Модулярность: 0.5609852653803823

Modularity

Gocha Gordienko

an Ousouvaliev

Elizaveta Bogaty
Talia Proutkin

Griamir Brachinskoi
Irina Elizaveta Oleinikova
Miroslava
Artem
Lilia
Anastasia Novgorodskaya
Yanina
Dasha
Krasnykh-Kitova
Natalia
Sergiy
Elizabeth Baskakova

Sacha Soldatov

Эта метрика мне не очень понравилась, 10 сообществ это слишком

```
In [67]: communities = set(partition.values())
communities_dict = {c: [k for k, v in partition.items() if v == c] for c
in communities}
highest_degree = {k: sorted(v, key=lambda x: graph.degree(x))[-5:] for k
, v in communities_dict.items()}
print('Количество сообществ: ', len(highest_degree))
print('Количество элементов в выделенных сообществах:', ', '.join([str(len(highest_degree[key])) for key in highest_degree]))
```

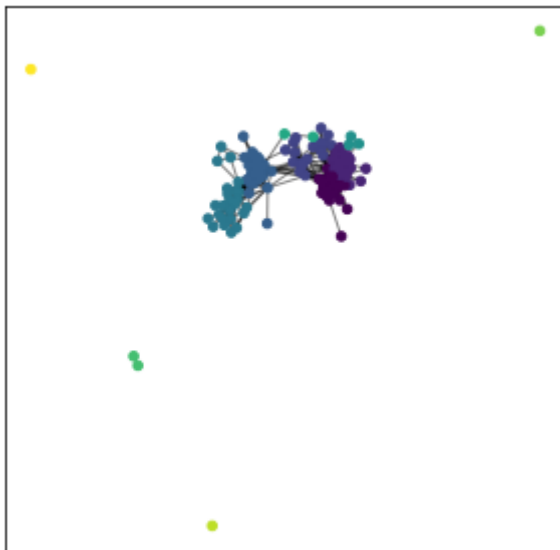
Количество сообществ: 11

Количество элементов в выделенных сообществах: 5, 5, 5, 5, 5, 3, 2, 2, 1, 1, 1

Тоже не очень нравится

```
In [68]: pos = nx.spring_layout(graph)
plt.figure(figsize=(5, 5))

nx.draw_networkx_nodes(graph, pos, partition.keys(), node_size=20, node_color=list(partition.values()))
nx.draw_networkx_edges(graph, pos, alpha=0.5)
plt.show()
```



```
In [69]: graph1 = nx.from_pandas_edgelist(clean_labels, 'Source', 'Target')
```

```
In [70]: #убираю сообщество из 2 человек ,чтобы посмотреть на диаметр оставшегося графа
graph1.remove_node('Elizaveta Bogatyriova')
graph1.remove_node('Talía Proutkina')
```

```
In [72]: cm = nxcom.girvan_newman(graph1)
communities = next(cm)
print(len(communities))
```

2

```
In [73]: nx.diameter(graph1) #6 рукопожатий?
```

Out[73]: 6

```
In [74]: nx.transitivity(graph1) #кластеризация
```

Out[74]: 0.5174135979223546

```
In [75]: nx.transitivity(graph)
```

Out[75]: 0.5174135979223546

```
In [78]: nx.average_clustering(graph1) #кластерный коэффициент
```

Out[78]: 0.618866436510218

```
In [79]: nx.average_clustering(graph)
```

Out[79]: 0.5957744052971502

Вконтакте я сейчас провожу мало времени, с друзьями из магистратуры общение перешло в телеграм, а преподавателями с филфака часто общаюсь в фейсбуке, так что не могу сказать, что граф прямо очень отображает ситуацию сейчас, но основные моменты жизни отразил, поэтому нормально. А гефи у меня завис и не прогрузился, так что без него.