

## Выбор алгоритма.

В качестве библиотеки для визуализации графов я выбрала NetworkX. Библиотека хороша тем, что содержит большой выбор методов визуализации с настраиваемыми параметрами. Также с библиотекой легко работать из Python.

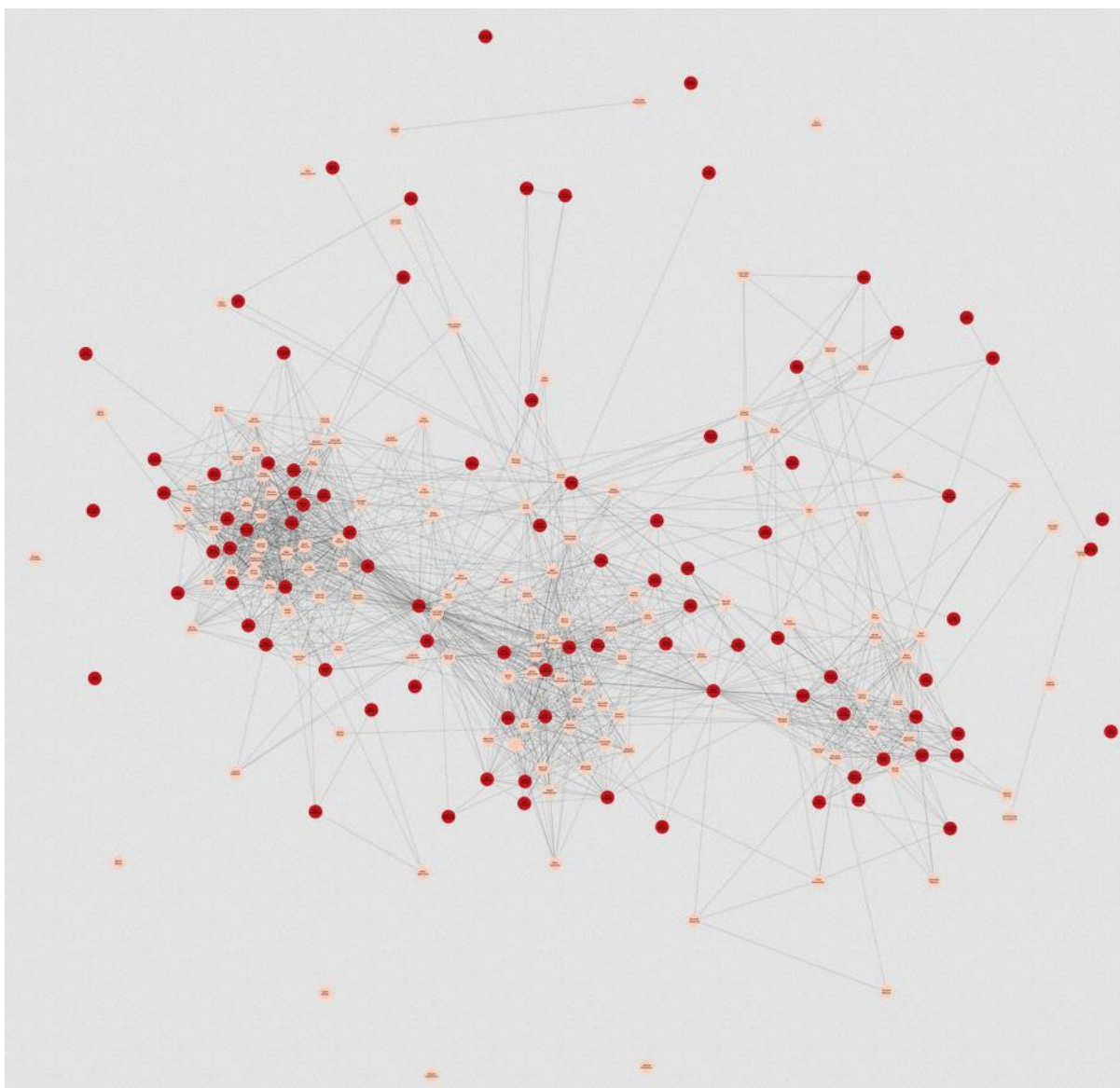
В качестве основы для визуализации я выбрала алгоритм Random Geometric Graph, но модифицировала его для работы со своими данными. Основная его функция - расположить сильно связанные компоненты вершин ближе друг к другу и сделать приемлемое расстояние между ребрами. Функции отрисовки самих вершин и ребер стандартные, как и в других алгоритмах. Они могут принимать на вход следующие параметры: сам граф, позиция вершин, прозрачность вершин, их цвета, размер, диапазон цветов, цветовая карта, и многие другие.

## Реализация алгоритма.

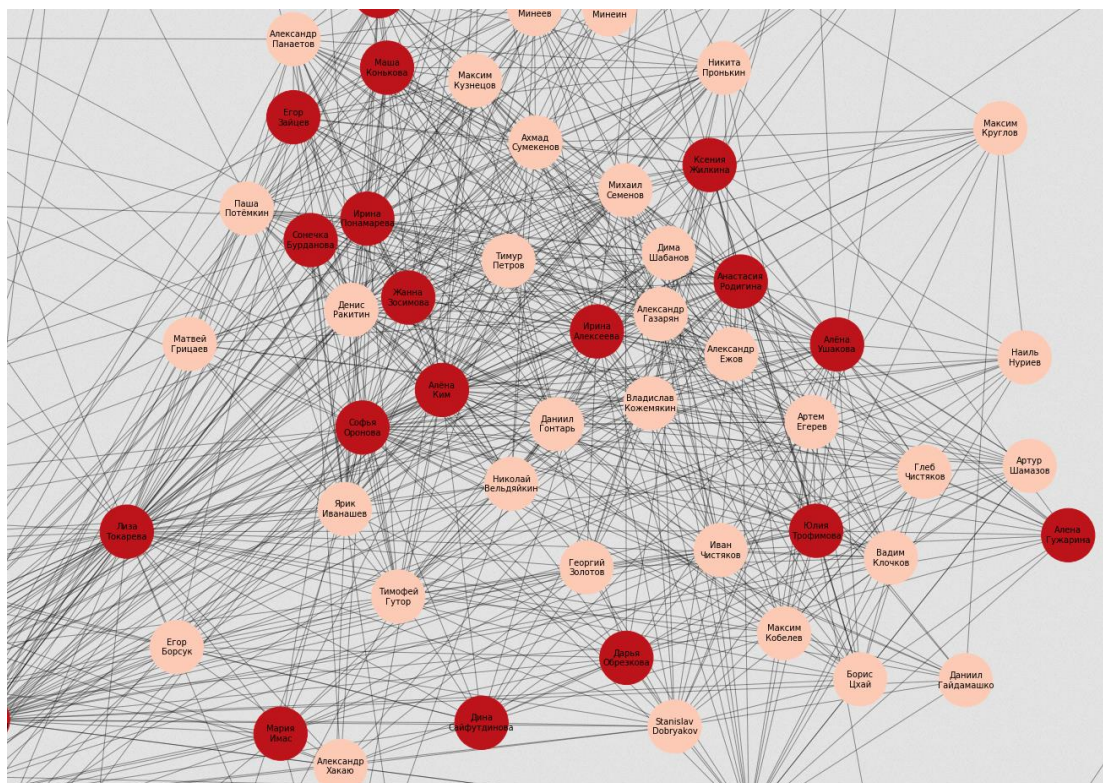
Модификации:

1. В качестве графа мной был передан не рандомно заданный граф, а сохраненные данные - вершины представляли собой моих друзей ВКонтакте (исключая меня).
2. Далее применялся `nx.spring_layout()` для позиционирования вершин на иллюстрации (вместо позиционирования для рандомного графа).
3. В граф передавались лейблы для каждой вершины, а именно - имя и фамилия друга.
4. Между вершинами, которые также являются друзьями друг друга, строились ребра, опираясь на информацию в матрице смежности, которую мы строили ранее (все ребра неориентированные).
5. Размер полотна для прорисовки графа пришлось увеличить, чтобы уместились все друзья.
6. Также увеличились и размеры самих нод.
7. В качестве цветов передавалась информация о гендере друзей, и вершины раскрашивались в один из двух цветов в отличие от него (ВКонтакте позволяет выбрать только один из двух полов). В функцию отрисовки нод я также передала пределы цветового диапазона, которые не были указаны ранее, чтобы обозначить оттенок цветов.
8. Границы полотна были немного сдвинуты для вмещения всего графика. Для их определения я рассмотрела массив позиционирования и его данные.

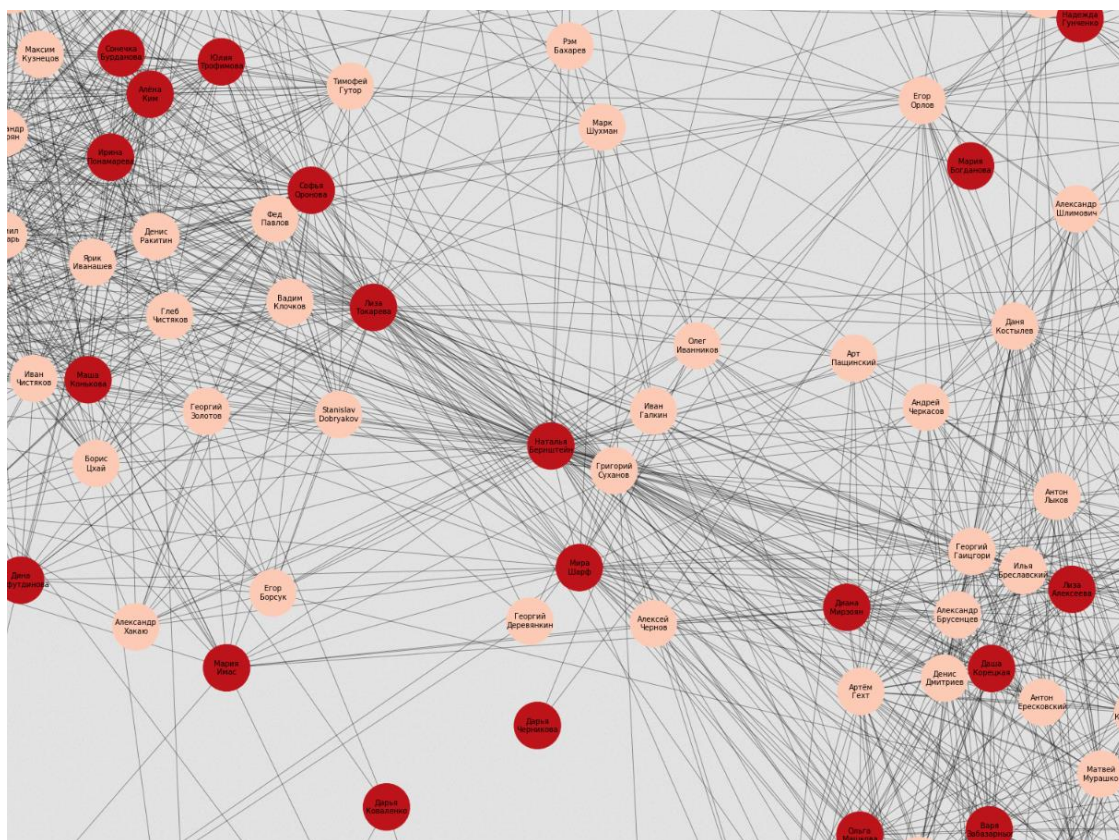
Подстроив алгоритм под свои данные, я получила следующий граф связей (чтобы рассмотреть поближе, перейдите в гитхаб в `pic1`, однако его размер слишком большой и придется проскроллить, но там отчетливо видно все вершины и лейблы на них):



Можно заметить, что у нас получился очень информативный график. Например, сразу заметно три полноценных кластера, сильно связанных внутри между собой. Если посмотреть на людей, сразу понятно, почему сложилась такая картина. Это кластер моих одноклассников первой школы, второй школы и ФКН. Увеличу последний кластер:



Люди, учащиеся в одном месте, знакомы друг с другом и имеют множество связей. Также можно заметить, что есть вершины между кластерами, однако тесно связанные сразу с двумя:

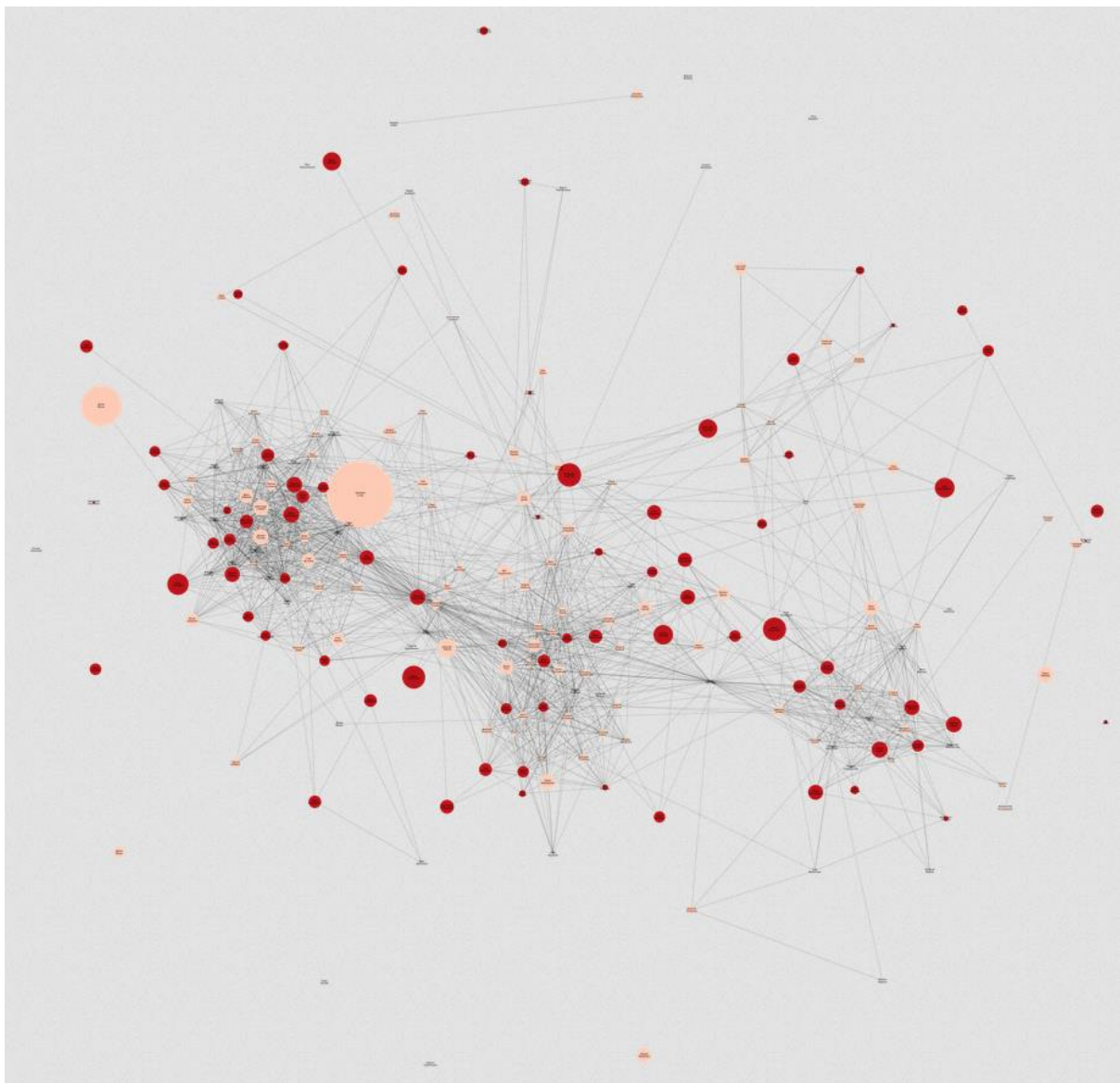




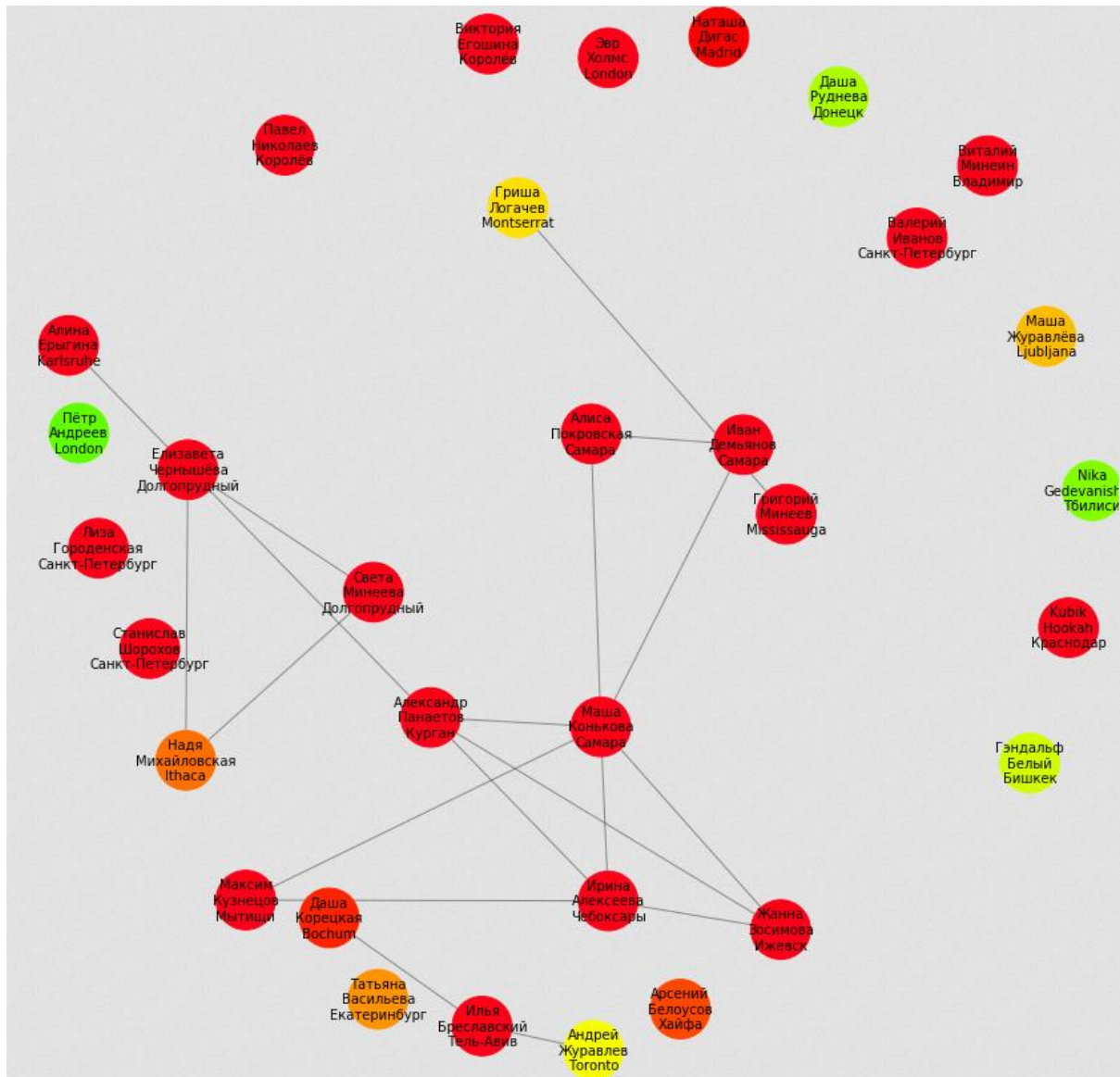
Наталья Бернштейн и другие подобные вершины училась со мной в одной школе, а после поступили со мной на ФКН. Даже не посвященный человек может догадаться об этом, лишь взглянув на график.

Есть также кластера поменьше - компании поменьше, много изолированных вершин - случайные знакомые или дальние родственники, вершины, имеющие огромное количество связей - мои близкие друзья, которых я со всеми познакомила, или близкие родственники. Граф получился очень информативным. Однако, я решила не останавливаться на этом и построить еще несколько, получив новую информацию.

На следующем графе добавлена новая информация. Теперь размер каждой ноды пропорционален количеству подписчиков конкретного человека (эту информацию я ранее извлекла при сборе данных). Для построения этого графика я передавала в параметр "размер вершин" новые значения. А именно - целый массив из количества подписчиков каждого человека (граф можно посмотреть более приближенно на гитхабе - [pic2](#)):



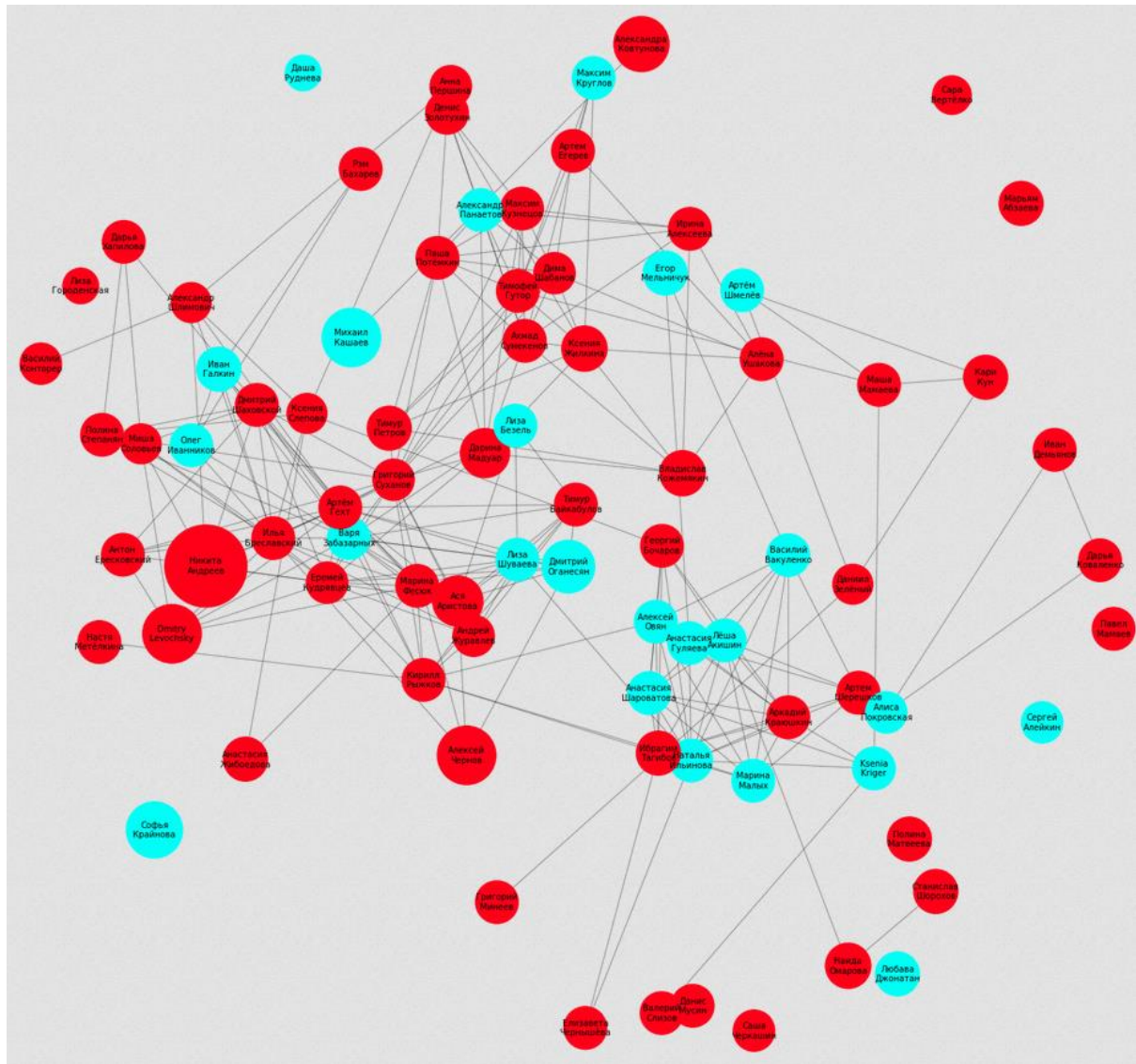
Далее я решила модифицировать алгоритм снова, чтобы извлечь новую информацию. На этот раз я захотела проверить, сколько моих друзей выбрали в качестве города в личной информации не Москву (я сама из Москвы, и большинство моих друзей и знакомых оттуда же). Для этого я отобрала только ноды, которые в графе “город” указали не Москву (и не проигнорировали это поле). Также я окрасила их в разные цвета в зависимости от выбранной страны. Получился следующий граф (pic4 на гитхабе):



Некоторые неточности возникают из-за того, что люди выбирают несколько городов проживания, но только одну страну. Эта погрешность небольшая. Однако, граф дал информацию о моих друзьях из других городов (их оказалось немало) и связей между ними. Основной кластер - друзья, учащиеся на ФКН, которые приехали из других городов.

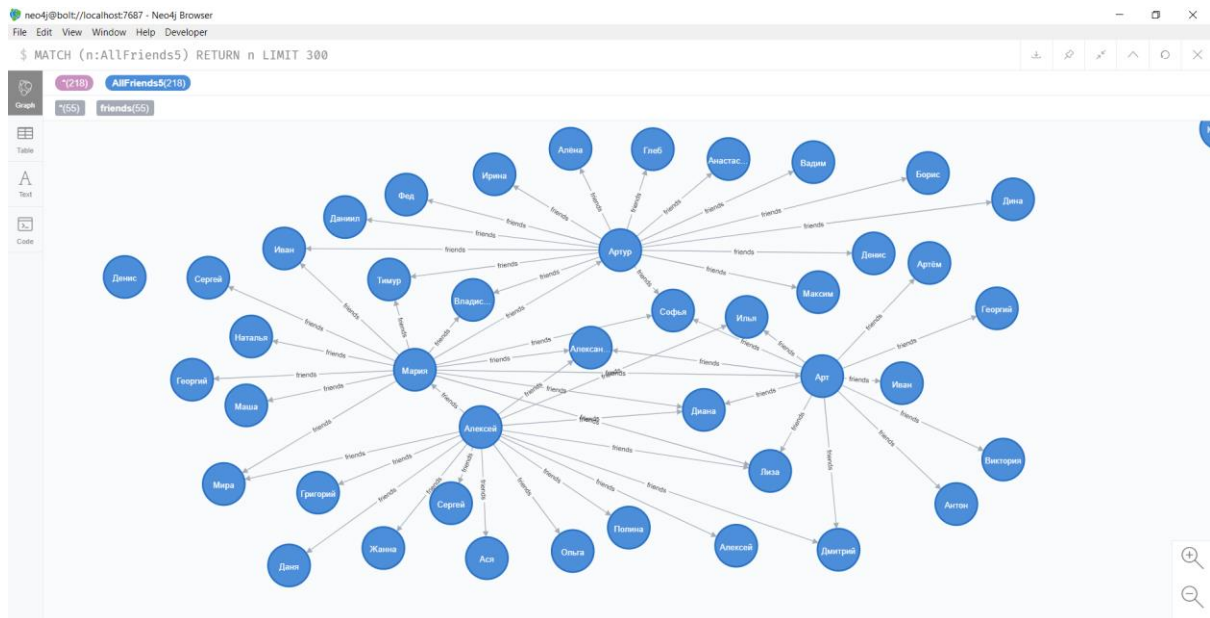
Для еще одного графа я рассмотрела даты рождения моих друзей. И посчитала примерно их возраст на 20.03.2020. Далее передала этот параметр как размер

вершин. Но разброс в возрасте получился очень мал, поэтому для превращения возраста в значения размера вершин использовала степенную функцию. Чтобы разнообразить граф, в качестве цветов передавался еще один параметр - оригинальность ника. Если человек задумывался над своим ником и специально вводил это поле ВКонтакте в настройках, то его цвет - красный. Если же человек воспользовался ником по умолчанию (айдишник), то его цвет - голубой. Это было сделано для получения дополнительной информации о базе данных моих друзей. Полученный граф (pic3 в гитхабе):



Таким образом, изначально модифицировав алгоритм, я вносила некоторые изменения каждый раз для нового графа, передавала новые параметры, которые сама доставала из базы данных и обрабатывала для лучшей визуализации.

Далее я решила построить графики в neo4j, чтобы интерактивно в нем взаимодействовать и добавлять связи. Я загрузила информацию о друзьях и соединила из них те, что являются друзьями друг друга (из-за большого объема данных и связей предоставляю только часть графика).



Приложение многофункционально, и имеет множество путей для улучшения графа. Например, можно добавить и другие связи между людьми (например, из одного города. Но на деле это слабо реализуемо, так как количество связей тогда очень велико и не наглядно, что плохо для визуализации). Также можно изменять цвета и размеры нод для лучшего представления. В целом, neo4j очень приятный для взаимодействия и строит удобные интерактивные визуализации.

Я построила несколько графов, модифицировав алгоритмы известных библиотек для своих данных и получила много наглядной информации благодаря визуализациям. А также поработала в приложении neo4j и построила интерактивную визуализацию графов.