Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Волгоградский государственный технический университет» (ВолгГТУ)

Кафедра САПРиПК

Контрольная работа

по дисциплине «Системы поддержки принятия решений»

Тема: «Автоматизированная оценка обеспеченности территории учебными заведениями и формирование графа сети учебных заведений города для ряда городов»

Выполнили:

студенты гр. ИВТ-365

Бесчастнова А.Ю.,

Громова С.А.

Проверил:

Парыгин Д.С.

Волгоград, 2022 г.

Цель работы

Разработка программы, позволяющей проанализировать обеспеченность ряда городов России учебными заведениями и визуализировать информацию об обеспеченности на графе.

Используемые компоненты

Язык программирования Python версии 3.7.2 для реализации программы.

Библиотека osmnx – библиотека, предоставляющая возможность использовать данные открытой карты OpenStreetMap для выбранных городов (Краснодар, Ростов-на-Дону, Волгоград).

Библиотека networkx – библиотека, предоставляющая возможность работы с графами.

Библиотека matplotlib – библиотека для вывода графиков, гистограмм, точек на карте.

Ход работы

Этап 1. Сбор всей информации об образовательных учреждениях.

1. Вычленение из карты информации об образовательных учреждениях города по их типу (тегу).
2. Запись обработанных данных (наименование, координаты, вычисленная площадь) о каждом образовательном учреждении в массив данных, в соответствующий типу образовательного учреждения подмассив.
3. Внесение всех данных в итоговый массив и вывод информации в виде текста. В пункте «Результаты работы программы» скриншот текстового вывода не фигурирует, т.к. вся необходимая информация полностью отображается на гистограммах. Текстовый вывод был нужен для сверки правильности отображаемых гистограмм.
4. Вывод информации в виде наглядных гистограмм.

Этап 2. Вывод карт городов с отмеченными на них точками различных цветов учреждений образования (цвета соответствуют тем, что использовались в гистограмме с количеством образовательных учреждений каждого типа на этапе 1).

1. Вычленение широт и долгот в раздельные массивы из общего массива (для удобства).
2. Нанесение точек на карту.

Этап 3. Вывод графа сети технических вузов Волгограда и ближайших к каждому из них пяти школ.

1. Для каждого вуза и каждой школы вычисляются расстояния между ними по теореме Пифагора: находятся разницы между широтами и долготами школы и вуза, данные разница являются катетами прямоугольного треугольника, а прямое расстояние между школой и вузом – это гипотенуза. Вычисленные пути записываются.
2. Сортируется массив путей по возрастанию, чтобы школы, соответствующие первым пяти кратчайшим путям, были внесены в новый массив для дальнейшего вывода на графе.
3. Построение графа на основе полученных данных.
4. Отдельное построение подграфа для ВолгГТУ и филиала ВолгГТУ, т.к. эти два вуза расположены очень близко и имеют одинаковый топ-5 школ. На графе из-за такой близости объектов их сложно различить (они сливаются).

Код программы (с подробными комментариями)

import numpy as np

import matplotlib.pyplot as plt

import osmnx as ox

import networkx as nx

cities=['Краснодар, Краснодарский край, Российская Федерация','Ростов-на-Дону, Ростовская область, Российская Федерация','Волгоград, Волгоградская область, Российская Федерация']#названия городов

c\_square=[294000000.0,348500000.0,859000000.0] #площади городов в кв.м.

edu\_square=[0.,0.,0.]#общая площадь образовательных учреждений городов

final\_mass=[]#массив с финальной обработанной информацией о городах

fin\_names=[]#массив наименований образ.учреждений

def square(mass): # расчет площади здания

pos = 0.

neg = 0.

for i in range(len(mass)):

if i != (len(mass) - 1):

pos += mass[i][0] \* mass[i + 1][1]

if i >= 1:

neg -= mass[i][0] \* mass[i - 1][1]

else:

pos += mass[i][0] \* mass[0][1]

neg -= mass[i][0] \* mass[i - 1][1]

neg -= mass[0][0] \* mass[len(mass) - 1][1]

s = abs(pos + neg) \* 10 \*\* 10

return s

for c\_num in range(len(cities)):

fin\_mass=[]#массив с готовой обработанной информацией о городе

#вычленение из карты информации об образовательных учреждениях города по их типу (тегу)

kindergartens = ox.geometries\_from\_place(cities[c\_num], tags={'amenity': 'kindergarten'})

schools = ox.geometries\_from\_place(cities[c\_num], tags={'amenity': 'school'})

colleges = ox.geometries\_from\_place(cities[c\_num], tags={'amenity': 'college'})

universities = ox.geometries\_from\_place(cities[c\_num], tags={'amenity': 'university'})

edu\_mass=[]#массив образовательных учреждений каждого города

edu\_mass.append(kindergartens)

edu\_mass.append(schools)

edu\_mass.append(colleges)

edu\_mass.append(universities)

final\_edu\_places = 0#все образовательные учреждения города (кол-во)

count\_amenities = [0,0,0,0] #количество учреждений каждого типа образования

fin\_edu\_coords=[]#массив с финальными координатами каждого учреждения образ-я

fin\_names=[]#массив наименований учреждений образ-я

for i in range(len(edu\_mass)): #для каждого типа образовательных учреждений

mass = [] #массив для промежуточной записи пар координат

names = [] # массив наименований образовательных учреждений

edu\_coords=[]#массив координат образ.учреждений города

edu\_sqs = [] # массив суммарной площади образовательных учреждений каждого типа

for item in edu\_mass[i].name: # запись наименования образ. учреждения в массив

names.append(item)

fin\_names.append(names)

for item in edu\_mass[i].geometry: # вычленение координат территории образ.учреждений

try:

main = str(item)

main = main.replace("POLYGON ((", "")

main = main.replace("))", "")

main = main.replace(",", "")

main = main.split(' ')

main = list(map(float, main))

except:

continue

count = 0

edu\_points=[]#промежуточный массив для записи точек образ. учреждений

while count < len(main):

mass.append(main[count])#широта

mass.append(main[count + 1])#долгота

edu\_points.append(mass)#добавление вычлененной точки в массив точек

mass = []#обнуление массива

count += 2#переход к следующей паре координат

sq = square(edu\_points) # подсчет суммы площадей учреждений каждого типа

edu\_sqs.append(sq) # присоединение значения к массиву площадей

edu\_coords.append(edu\_points[0]) # добавление точки от здания

edu\_points=[]#очистка массива для новых значений

fin\_edu\_coords.append(edu\_coords)#присоединение координат к массиву со всеми координатами учреждений

for j in range(len(edu\_sqs)):#подсчет сумм всех учреждений, всей площади, учреждений по типам

final\_edu\_places+=1

edu\_square[c\_num] += edu\_sqs[j]

count\_amenities[i]+=1

#внесение в итоговый массив всех значений

fin\_mass.append(cities[c\_num]) # название города

fin\_mass.append(final\_edu\_places) # общее количество образовательных учреждений

fin\_mass.append(count\_amenities)# кол-во учреждений по типам

fin\_mass.append(edu\_square[c\_num])# площадь всех образовательных учреждений

fin\_mass.append((edu\_square[c\_num]/c\_square[c\_num])\*100)#процент площади образ.учреждений

fin\_mass.append(fin\_names)#названия образовательных учреждений

fin\_mass.append(fin\_edu\_coords)#точки образовательных учреждений города

#добавление инфо о городе в финальный массив

final\_mass.append(fin\_mass)

# вывод результирующего массива (текстовый вывод)

print("Город:",cities[c\_num])

print("Сумма образовательных учреждений:",str(final\_edu\_places))

print(" Из них\n - детских садов:",str(count\_amenities[0]))

print(" - школ:",str(count\_amenities[1]))

print(" - колледжей:",str(count\_amenities[2]))

print(" - университетов:",str(count\_amenities[3]))

print("Площадь всех образовательных учреждений: {0:.3f} кв.м.".format(edu\_square[c\_num]))

print("Процент занимаемой образ. учреждениями площади города: {0:.2f} %\n".format((edu\_square[c\_num]/c\_square[c\_num])\*100))

#вывод графиков, статистики

#Процент площадей образ.учреждений в каждом городе

index = np.arange(3)

values = [final\_mass[0][4], final\_mass[1][4], final\_mass[2][4]]#значения для вывода

plt.bar(index, values)#из чего строится столбчатая гистограмма

for i in range(len(final\_mass)):

plt.text(i, final\_mass[i][4]/2, str("{0:.3f} кв.км.\n({1:.2f}%)".format(final\_mass[i][3]/1000000.,final\_mass[i][4])), ha = 'center')

plt.xticks(index, ['Краснодар', 'Ростов-на-Дону', 'Волгоград'])#подписи по иксу

plt.title("Процент занимаемой образовательными \nучреждениями площади городов") #название гистограммы

plt.show()#отображение гистограммы

#Кол-во образ.учреждений каждого уровня образ-я в каждом городе

index = np.arange(3)

values1 = np.array([final\_mass[0][2][0], final\_mass[1][2][0], final\_mass[2][2][0]])#дет.сады

values2 = np.array([final\_mass[0][2][1], final\_mass[1][2][1], final\_mass[2][2][1]])#школы

values3 = np.array([final\_mass[0][2][2], final\_mass[1][2][2], final\_mass[2][2][2]])#колледжи

values4 = np.array([final\_mass[0][2][3], final\_mass[1][2][3], final\_mass[2][2][3]])#вузы

plt.bar(index,values1,color='g',label='дошкольное')

plt.bar(index,values2,color='c',bottom=values1,label='среднее общее/полное')

plt.bar(index,values3,color='b',bottom=(values2+values1),label='среднее профессиональное')

plt.bar(index,values4,color='m',bottom=(values3+values2+values1),label='высшее профессиональное')

summ=[0,0,0]

for i in range(len(final\_mass)):

for j in range(len(final\_mass[i][2])):

summ[i]+=final\_mass[i][2][j]

plt.text(i, summ[i]-final\_mass[i][2][j]+final\_mass[i][2][j]/2, final\_mass[i][2][j], ha = 'center')

plt.xticks(index, ['Краснодар', 'Ростов-на-Дону', 'Волгоград'])#подписи по иксу

plt.title("Кол-во образовательных учреждений \nкаждого уровня образования в каждом городе") #название гистограммы

plt.ylim(0,650)

plt.legend(loc=2)

plt.show()#отображение гистограммы

#вывод карты города с отмеченными на ней точками различных цветов учреждений образования

for c in range(len(cities)):

print("\nГород:",cities[c])

G = ox.graph\_from\_place(cities[c],network\_type="walk")

fig, ax = ox.plot\_graph(G,figsize=(25,25), close=False, show=False)

#вычленение широт и долгот в раздельные массивы (сначала долгота, потом широта)

latitudes=[]

longitudes=[]

for q in range(len(final\_mass[c][6])):

lats=[]

lons=[]

for r in range(len(final\_mass[c][6][q])):

lats.append(final\_mass[c][6][q][r][1])

lons.append(final\_mass[c][6][q][r][0])

latitudes.append(lats)

longitudes.append(lons)

ax.scatter(longitudes[0],latitudes[0], c='g',s=50)#сады

ax.scatter(longitudes[1],latitudes[1], c='c',s=75)#школы

ax.scatter(longitudes[2],latitudes[2], c='b',s=100)#колледжи

ax.scatter(longitudes[3],latitudes[3], c='m',s=125)#вузы

plt.show()

#среди всех школ найдем ближайшие через теорему Пифагора

vuz\_names=['Технический университет (ВолгГТУ)','ФГБОУ ВО ВолгГТУ "Кировский вечерний факультет"','Волгоградский государственный технический университет (филиал)','Институт архитектуры и строительства ВолгГТУ']

vuz\_coords=[[44.5282608, 48.7145228],[44.421855, 48.6433659],[44.530549, 48.714677],[44.497759, 48.693182]]

nearest\_coords=[]#координаты ближайших школ

nearest\_schools=[]#

for o in range(len(vuz\_names)):

delta\_lat=[]#разница между широтой школы и широтой вуза

delta\_lon=[]#разница между долготой школы и долготой вуза

route=[]#расстояния напрямую от школы до вуза (прямые пути)

nears\_coords=[]#

nears\_names=[]#

for a in range(len(final\_mass[2][6][1])):

delta\_lat.append(abs(final\_mass[2][6][1][a][0]-vuz\_coords[o][0]))

delta\_lon.append(abs(final\_mass[2][6][1][a][1]-vuz\_coords[o][1]))

route.append((delta\_lat[a]\*\*2+delta\_lon[a]\*\*2)\*\*0.5)#дельта широт и дельта долгот - катеты, путь - гипотенуза

#сортировка прямых путей в порядке возрастания, чтобы наименьшие 5 записать в другой массив

route\_ind=[]#индексы

for d in range(len(final\_mass[2][6][1])):

route\_ind.append(d)

for b in range(len(final\_mass[2][6][1])):

for c in range(len(final\_mass[2][6][1])):

if (route[b]<route[c]):

buf=route[b]

route[b]=route[c]

route[c]=buf

buf=route\_ind[b]

route\_ind[b]=route\_ind[c]

route\_ind[c]=buf

for e in range(5):

nears=[]

near\_names=[]

nears.append(final\_mass[2][6][1][route\_ind[e]][0])

nears.append(final\_mass[2][6][1][route\_ind[e]][1])

nears\_coords.append(nears)

near\_names.append(final\_mass[2][5][1][route\_ind[e]])

nears\_names.append(near\_names)

nearest\_schools.append(nears\_names)

nearest\_coords.append(nears\_coords)

for q in range(len(vuz\_names)):

print("\nТоп-5 школ, ближайших к:",vuz\_names[q], vuz\_coords[q])

for w in range(5):

print(w+1,nearest\_schools[q][w],nearest\_coords[q][w], "длина пути: "+str(route[w]))

#построение графа: для каждого технического вуза Волгограда (их 4) построить маршруты к 5 ближайшим к ним школам

G = nx.Graph()

#вузы

G.add\_node('VSTU',title="VSTU", size=500)

G.add\_node('Kirov.VF',title="Kirov.VF", size=500)

G.add\_node('VSTU fil.',title="VSTU fil.", size=500)

G.add\_node('Archit.i stroitel.',title="Archit.i stroitel.", size=500)

#школы

G.add\_node('school\_96', title="school\_96", size=100)

G.add\_node('school\_61', title="school\_61", size=100)

G.add\_node('internat\_4', title="internat\_4", size=100)

G.add\_node('school\_120', title="school\_120", size=100)

G.add\_node('kazak\_corp', title="kazak\_corp", size=100)

G.add\_node('school\_23', title="school\_23", size=100)

G.add\_node('sr\_school\_3', title="sr\_school\_3", size=100)

G.add\_node('kr\_school\_4', title="kr\_school\_4", size=100)

G.add\_node('noname1', title="noname1", size=100)

G.add\_node('sozvezdie', title="sozvezdie", size=100)

G.add\_node('school\_96', title="school\_96", size=100)

G.add\_node('school\_61', title="school\_61", size=100)

G.add\_node('internat\_4', title="internat\_4", size=100)

G.add\_node('kazak\_corp', title="kazak\_corp", size=100)

G.add\_node('school\_120', title="school\_120", size=100)

G.add\_node('school\_64', title="school\_64", size=100)

G.add\_node('school\_86', title="school\_86", size=100)

G.add\_node('school\_94', title="school\_94", size=100)

G.add\_node('gimnasia\_15', title="gimnasia\_15", size=100)

G.add\_node('school\_12', title="school\_12", size=100)

node\_labelss = nx.get\_node\_attributes(G, 'title') # не трогать

sizeee = nx.get\_node\_attributes(G, 'size')

squareee = nx.get\_node\_attributes(G, 'square')

G.add\_edge('Kirov.VF','Archit.i stroitel.',title='17.1км')

G.add\_edge('VSTU','Archit.i stroitel.',title='4.2км')

G.add\_edge('VSTU','VSTU fil.',title='0.14км')

#G.add\_edge('Kirov.VF','VSTU fil.',title='20.2км')

#G.add\_edge('VSTU','Kirov.VF',title='20.6км')

#G.add\_edge('VSTU fil.','Archit.i stroitel.',title='4км')

G.add\_edge('VSTU','school\_96',title='1')

G.add\_edge('VSTU','school\_61',title='2')

G.add\_edge('VSTU','internat\_4',title='3')

G.add\_edge('VSTU','school\_120',title='4')

G.add\_edge('VSTU','kazak\_corp',title='5')

G.add\_edge('Kirov.VF','school\_23',title='1')

G.add\_edge('Kirov.VF','sr\_school\_3',title='2')

G.add\_edge('Kirov.VF','kr\_school\_4',title='3')

G.add\_edge('Kirov.VF','noname1',title='4')

G.add\_edge('Kirov.VF','sozvezdie',title='5')

G.add\_edge('VSTU fil.','school\_96',title='1')

G.add\_edge('VSTU fil.','school\_61',title='2')

G.add\_edge('VSTU fil.','internat\_4',title='3')

G.add\_edge('VSTU fil.','kazak\_corp',title='4')

G.add\_edge('VSTU fil.','school\_120',title='5')

G.add\_edge('Archit.i stroitel.','school\_64',title='1')

G.add\_edge('Archit.i stroitel.','school\_94',title='2')

G.add\_edge('Archit.i stroitel.','school\_86',title='3')

G.add\_edge('Archit.i stroitel.','gimnasia\_15',title='4')

G.add\_edge('Archit.i stroitel.','school\_12',title='5')

edge\_labels = nx.get\_edge\_attributes(G, 'title')

pos = {

"VSTU":(vuz\_coords[0][0], vuz\_coords[0][1]),#

"Kirov.VF":(vuz\_coords[1][0], vuz\_coords[1][1]),#

"VSTU fil.":(vuz\_coords[2][0], vuz\_coords[2][1]),#

"Archit.i stroitel.":(vuz\_coords[3][0], vuz\_coords[3][1]),#

"school\_96":(nearest\_coords[0][0][0],nearest\_coords[0][0][1]),

"school\_61":(nearest\_coords[0][1][0],nearest\_coords[0][1][1]),

"internat\_4":(nearest\_coords[0][2][0],nearest\_coords[0][2][1]),

"school\_120":(nearest\_coords[0][3][0],nearest\_coords[0][3][1]),

"kazak\_corp":(nearest\_coords[0][4][0],nearest\_coords[0][4][1]),

"school\_23":(nearest\_coords[1][0][0],nearest\_coords[1][0][1]),

"sr\_school\_3":(nearest\_coords[1][1][0],nearest\_coords[1][1][1]),

"kr\_school\_4":(nearest\_coords[1][2][0],nearest\_coords[1][2][1]),

"noname1":(nearest\_coords[1][3][0],nearest\_coords[1][3][1]),

"sozvezdie":(nearest\_coords[1][4][0],nearest\_coords[1][4][1]),

"school\_96":(nearest\_coords[2][0][0],nearest\_coords[2][0][1]),

"school\_61":(nearest\_coords[2][1][0],nearest\_coords[2][1][1]),

"internat\_4":(nearest\_coords[2][2][0],nearest\_coords[2][2][1]),

"kazak\_corp":(nearest\_coords[2][3][0],nearest\_coords[2][3][1]),

"school\_120":(nearest\_coords[2][4][0],nearest\_coords[2][4][1]),

"school\_64":(nearest\_coords[3][0][0],nearest\_coords[3][0][1]),

"school\_94":(nearest\_coords[3][1][0],nearest\_coords[3][1][1]),

"school\_86":(nearest\_coords[3][2][0],nearest\_coords[3][2][1]),

"gimnasia\_15":(nearest\_coords[3][3][0],nearest\_coords[3][3][1]),

"school\_12":(nearest\_coords[3][4][0],nearest\_coords[3][4][1]),

}

pos\_labels = {

"VSTU":(vuz\_coords[0][0], vuz\_coords[0][1]),#

"Kirov.VF":(vuz\_coords[1][0], vuz\_coords[1][1]),#

"VSTU fil.":(vuz\_coords[2][0], vuz\_coords[2][1]),#

"Archit.i stroitel.":(vuz\_coords[3][0], vuz\_coords[3][1]),#

"school\_96":(nearest\_coords[0][0][0],nearest\_coords[0][0][1]),

"school\_61":(nearest\_coords[0][1][0],nearest\_coords[0][1][1]),

"internat\_4":(nearest\_coords[0][2][0],nearest\_coords[0][2][1]),

"school\_120":(nearest\_coords[0][3][0],nearest\_coords[0][3][1]),

"kazak\_corp":(nearest\_coords[0][4][0],nearest\_coords[0][4][1]),

"school\_23":(nearest\_coords[1][0][0],nearest\_coords[1][0][1]),

"sr\_school\_3":(nearest\_coords[1][1][0],nearest\_coords[1][1][1]),

"kr\_school\_4":(nearest\_coords[1][2][0],nearest\_coords[1][2][1]),

"noname1":(nearest\_coords[1][3][0],nearest\_coords[1][3][1]),

"sozvezdie":(nearest\_coords[1][4][0],nearest\_coords[1][4][1]),

"school\_96":(nearest\_coords[2][0][0],nearest\_coords[2][0][1]),

"school\_61":(nearest\_coords[2][1][0],nearest\_coords[2][1][1]),

"internat\_4":(nearest\_coords[2][2][0],nearest\_coords[2][2][1]),

"kazak\_corp":(nearest\_coords[2][3][0],nearest\_coords[2][3][1]),

"school\_120":(nearest\_coords[2][4][0],nearest\_coords[2][4][1]),

"school\_64":(nearest\_coords[3][0][0],nearest\_coords[3][0][1]),

"school\_94":(nearest\_coords[3][1][0],nearest\_coords[3][1][1]),

"school\_86":(nearest\_coords[3][2][0],nearest\_coords[3][2][1]),

"gimnasia\_15":(nearest\_coords[3][3][0],nearest\_coords[3][3][1]),

"school\_12":(nearest\_coords[3][4][0],nearest\_coords[3][4][1]),

}

pos\_area = {

"VSTU":(vuz\_coords[0][0], vuz\_coords[0][1]),#

"Kirov.VF":(vuz\_coords[1][0], vuz\_coords[1][1]),#

"VSTU fil.":(vuz\_coords[2][0], vuz\_coords[2][1]),#

"Archit.i stroitel.":(vuz\_coords[3][0], vuz\_coords[3][1]),#

"school\_96":(nearest\_coords[0][0][0],nearest\_coords[0][0][1]),

"school\_61":(nearest\_coords[0][1][0],nearest\_coords[0][1][1]),

"internat\_4":(nearest\_coords[0][2][0],nearest\_coords[0][2][1]),

"school\_120":(nearest\_coords[0][3][0],nearest\_coords[0][3][1]),

"kazak\_corp":(nearest\_coords[0][4][0],nearest\_coords[0][4][1]),

"school\_23":(nearest\_coords[1][0][0],nearest\_coords[1][0][1]),

"sr\_school\_3":(nearest\_coords[1][1][0],nearest\_coords[1][1][1]),

"kr\_school\_4":(nearest\_coords[1][2][0],nearest\_coords[1][2][1]),

"noname1":(nearest\_coords[1][3][0],nearest\_coords[1][3][1]),

"sozvezdie":(nearest\_coords[1][4][0],nearest\_coords[1][4][1]),

"school\_96":(nearest\_coords[2][0][0],nearest\_coords[2][0][1]),

"school\_61":(nearest\_coords[2][1][0],nearest\_coords[2][1][1]),

"internat\_4":(nearest\_coords[2][2][0],nearest\_coords[2][2][1]),

"kazak\_corp":(nearest\_coords[2][3][0],nearest\_coords[2][3][1]),

"school\_120":(nearest\_coords[2][4][0],nearest\_coords[2][4][1]),

"school\_64":(nearest\_coords[3][0][0],nearest\_coords[3][0][1]),

"school\_94":(nearest\_coords[3][1][0],nearest\_coords[3][1][1]),

"school\_86":(nearest\_coords[3][2][0],nearest\_coords[3][2][1]),

"gimnasia\_15":(nearest\_coords[3][3][0],nearest\_coords[3][3][1]),

"school\_12":(nearest\_coords[3][4][0],nearest\_coords[3][4][1]),

}

nx.draw(G, pos=pos, node\_size=[v for v in sizeee.values()]) # заготовка, отрисовка без надписей

nx.draw\_networkx\_labels(G, pos=pos\_labels, labels=node\_labelss, font\_size=6,font\_weight='light') # за название отвечает

nx.draw\_networkx\_edge\_labels(G, pos=pos, edge\_labels=edge\_labels, font\_size=6) # вес между точками (вес дуг)

plt.margins(0.)

plt.show()

#более крупный граф, отобрадающий ближайшие школы к ВолгГТУ и филиалу ВолгГТУ

G = nx.Graph()

#вузы

G.add\_node('VSTU',title="VSTU", size=500)

#G.add\_node('Kirov.VF',title="Kirov.VF", size=500)

G.add\_node('VSTU fil.',title="VSTU fil.", size=500)

#G.add\_node('Archit.i stroitel.',title="Archit.i stroitel.", size=500)

#школы

G.add\_node('school\_96', title="school\_96", size=100)

G.add\_node('school\_61', title="school\_61", size=100)

G.add\_node('internat\_4', title="internat\_4", size=100)

G.add\_node('school\_120', title="school\_120", size=100)

G.add\_node('kazak\_corp', title="kazak\_corp", size=100)

node\_labelss = nx.get\_node\_attributes(G, 'title') # не трогать

sizeee = nx.get\_node\_attributes(G, 'size')

squareee = nx.get\_node\_attributes(G, 'square')

#G.add\_edge('Kirov.VF','Archit.i stroitel.',title='17.1км')

#G.add\_edge('VSTU','Archit.i stroitel.',title='4.2км')

G.add\_edge('VSTU','VSTU fil.',title='0.14км')

#G.add\_edge('Kirov.VF','VSTU fil.',title='20.2км')

#G.add\_edge('VSTU','Kirov.VF',title='20.6км')

#G.add\_edge('VSTU fil.','Archit.i stroitel.',title='4км')

G.add\_edge('VSTU','school\_96',title='1')

G.add\_edge('VSTU','school\_61',title='2')

G.add\_edge('VSTU','internat\_4',title='3')

G.add\_edge('VSTU','school\_120',title='4')

G.add\_edge('VSTU','kazak\_corp',title='5')

G.add\_edge('VSTU fil.','school\_96',title='1')

G.add\_edge('VSTU fil.','school\_61',title='2')

G.add\_edge('VSTU fil.','internat\_4',title='3')

G.add\_edge('VSTU fil.','kazak\_corp',title='4')

G.add\_edge('VSTU fil.','school\_120',title='5')

edge\_labels = nx.get\_edge\_attributes(G, 'title')

pos = {

"VSTU":(vuz\_coords[0][0], vuz\_coords[0][1]),#

#"Kirov.VF":(vuz\_coords[1][0], vuz\_coords[1][1]),#

"VSTU fil.":(vuz\_coords[2][0], vuz\_coords[2][1]),#

#"Archit.i stroitel.":(vuz\_coords[3][0], vuz\_coords[3][1]),#

"school\_96":(nearest\_coords[0][0][0],nearest\_coords[0][0][1]),

"school\_61":(nearest\_coords[0][1][0],nearest\_coords[0][1][1]),

"internat\_4":(nearest\_coords[0][2][0],nearest\_coords[0][2][1]),

"school\_120":(nearest\_coords[0][3][0],nearest\_coords[0][3][1]),

"kazak\_corp":(nearest\_coords[0][4][0],nearest\_coords[0][4][1]),

}

pos\_labels = {

"VSTU":(vuz\_coords[0][0], vuz\_coords[0][1]),#

#"Kirov.VF":(vuz\_coords[1][0], vuz\_coords[1][1]),#

"VSTU fil.":(vuz\_coords[2][0], vuz\_coords[2][1]),#

#"Archit.i stroitel.":(vuz\_coords[3][0], vuz\_coords[3][1]),#

"school\_96":(nearest\_coords[0][0][0],nearest\_coords[0][0][1]),

"school\_61":(nearest\_coords[0][1][0],nearest\_coords[0][1][1]),

"internat\_4":(nearest\_coords[0][2][0],nearest\_coords[0][2][1]),

"school\_120":(nearest\_coords[0][3][0],nearest\_coords[0][3][1]),

"kazak\_corp":(nearest\_coords[0][4][0],nearest\_coords[0][4][1]),

}

pos\_area = {

"VSTU":(vuz\_coords[0][0], vuz\_coords[0][1]),#

#"Kirov.VF":(vuz\_coords[1][0], vuz\_coords[1][1]),#

"VSTU fil.":(vuz\_coords[2][0], vuz\_coords[2][1]),#

#"Archit.i stroitel.":(vuz\_coords[3][0], vuz\_coords[3][1]),#

"school\_96":(nearest\_coords[0][0][0],nearest\_coords[0][0][1]),

"school\_61":(nearest\_coords[0][1][0],nearest\_coords[0][1][1]),

"internat\_4":(nearest\_coords[0][2][0],nearest\_coords[0][2][1]),

"school\_120":(nearest\_coords[0][3][0],nearest\_coords[0][3][1]),

"kazak\_corp":(nearest\_coords[0][4][0],nearest\_coords[0][4][1]),

}

nx.draw(G, pos=pos, node\_size=[v for v in sizeee.values()]) # заготовка, отрисовка без надписей

nx.draw\_networkx\_labels(G, pos=pos\_labels, labels=node\_labelss, font\_size=6,font\_weight='light') # за название отвечает

nx.draw\_networkx\_edge\_labels(G, pos=pos, edge\_labels=edge\_labels, font\_size=6) # вес между точками (вес дуг)

plt.margins(0.)

plt.show()

Результаты работы программы

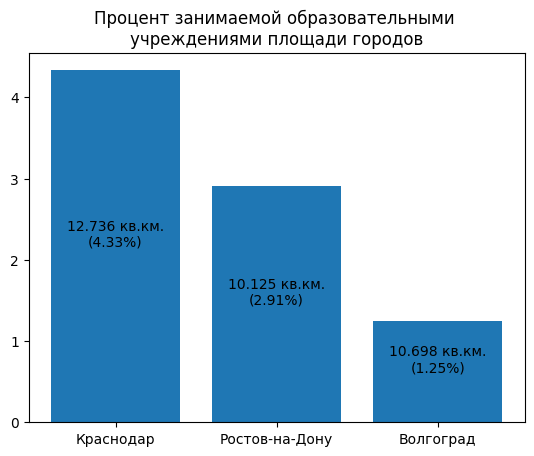


Рис. 1. Гистограмма «Процент занимаемой образовательными учреждениями площади городов»

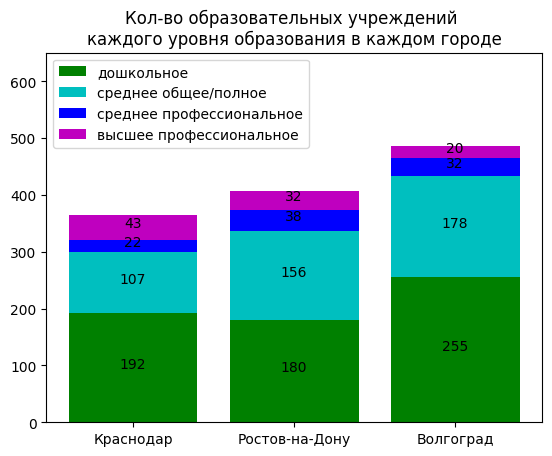


Рис. 2. Гистограмма «Количество образовательных учреждений каждого уровня образования в каждом городе»

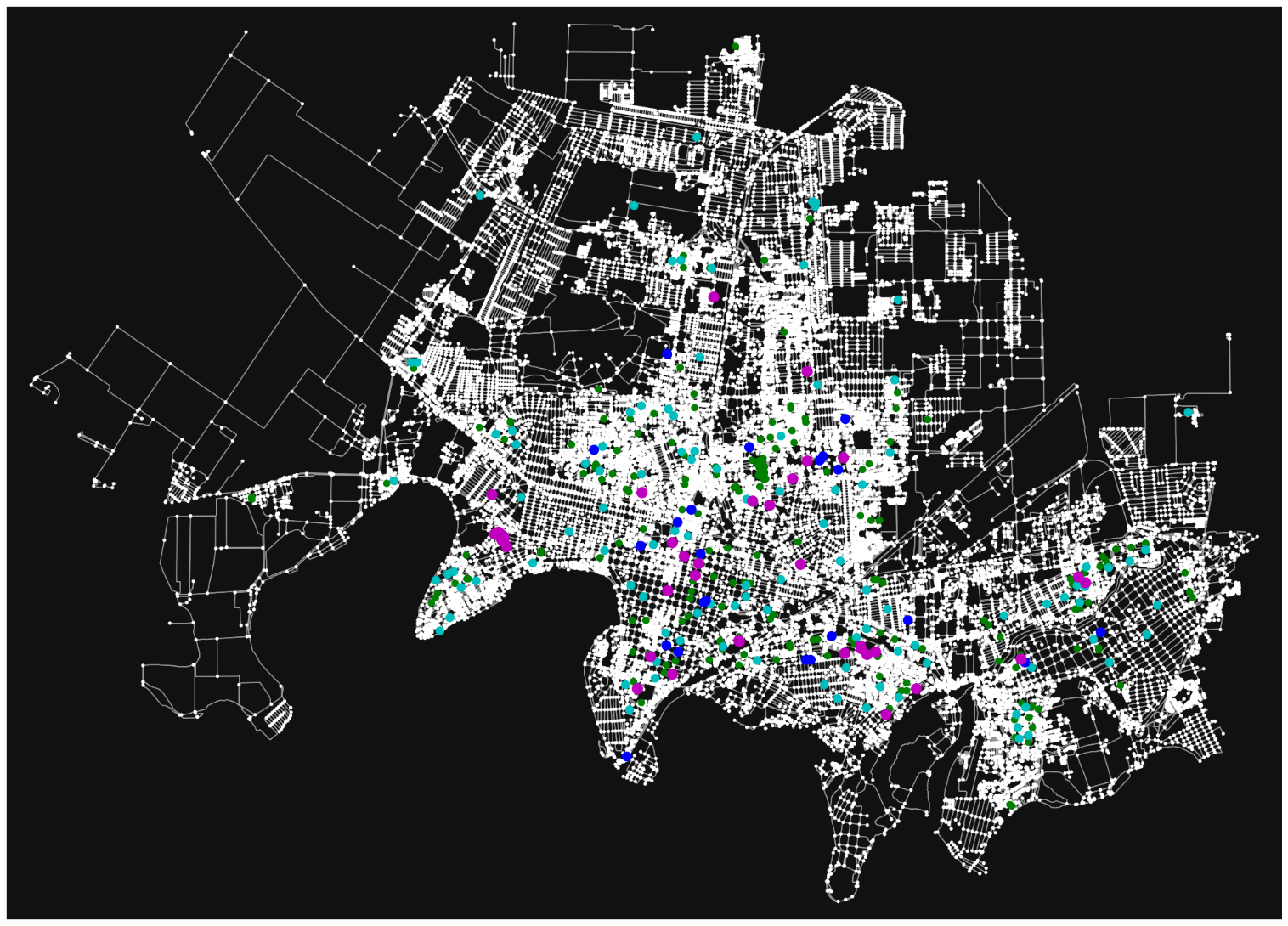


Рис. 3. Карта Краснодара с нанесенными точками образовательных учреждений каждого типа.



Рис. 4. Карта Ростова-на-Дону с нанесенными точками образовательных учреждений каждого типа.

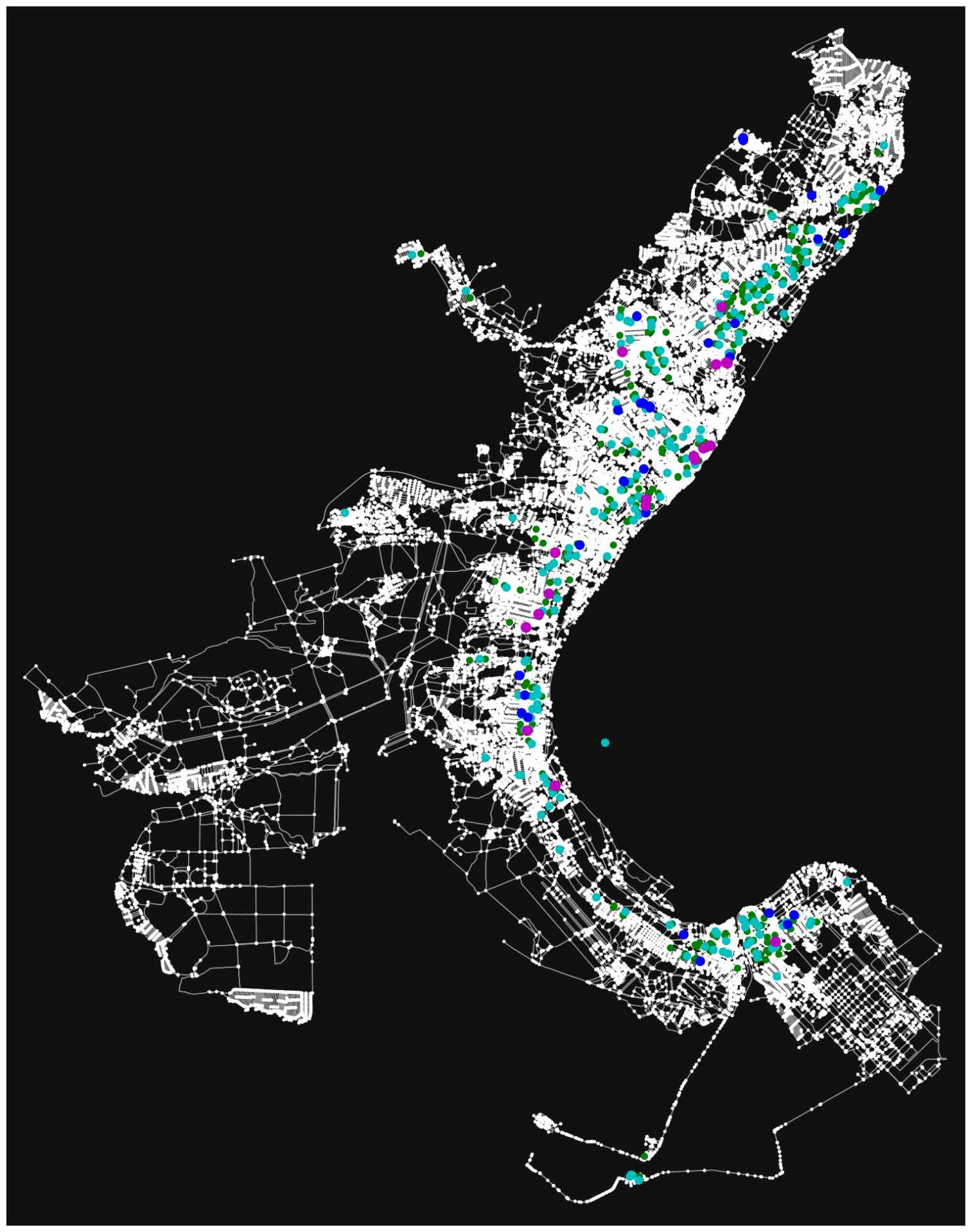


Рис. 5. Карта Волгограда с нанесенными точками образовательных учреждений каждого типа.

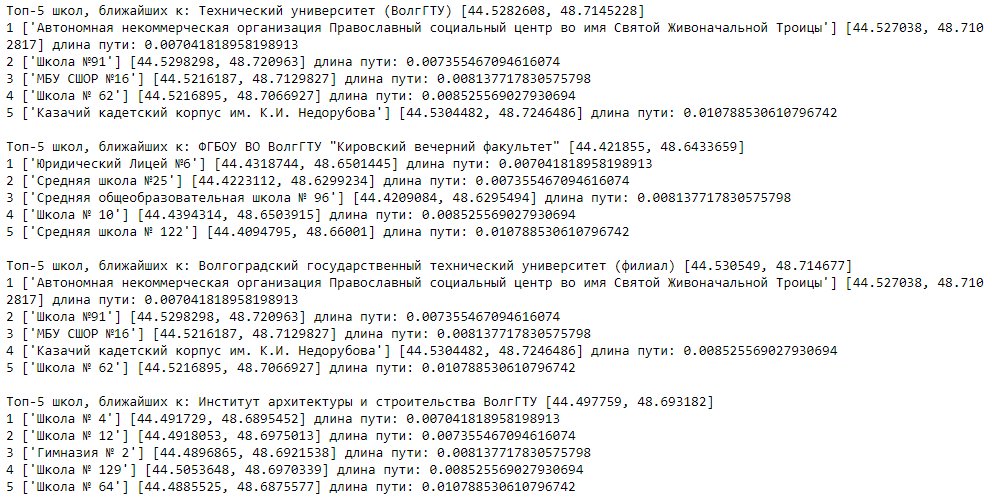


Рис. 6. Текстовый вывод информации о ближайших школах к каждому техническому вузу Волгограда.

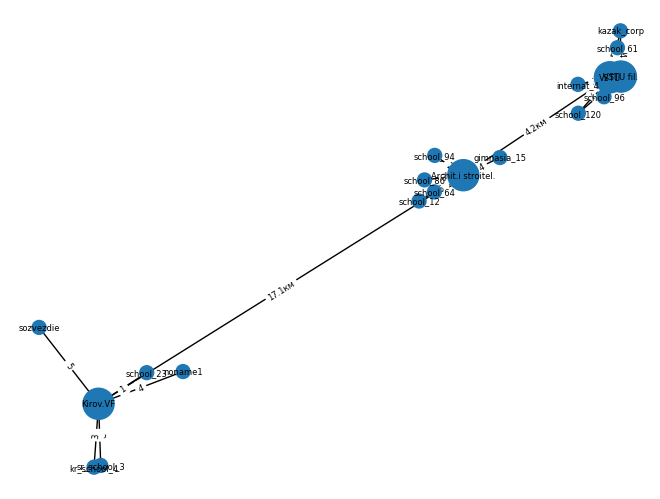
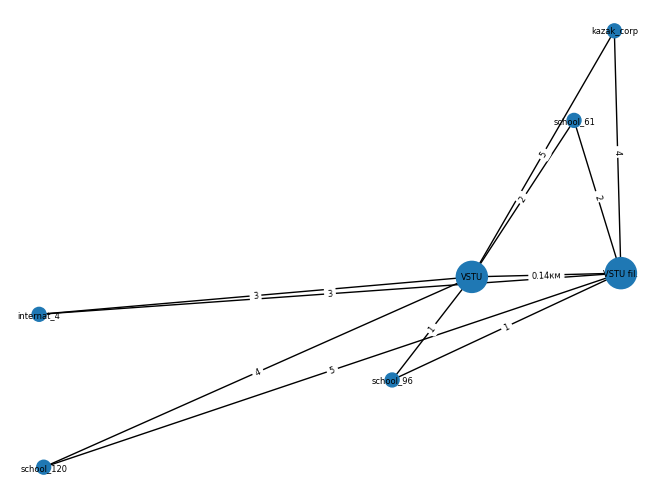


Рис. 7. Граф с сетью технических вузов и ближайших к ним школ для города Волгограда. Между вузами расстояние указано в километрах. Между школами и вузами указано место в топе по близости.

Рис. 8. Фрагмент графа с рис. 7, отображающий ближайшие школы к ВолгГТУ и филиалу ВолгГТУ. Между школами и вузами указано место в топе по близости.