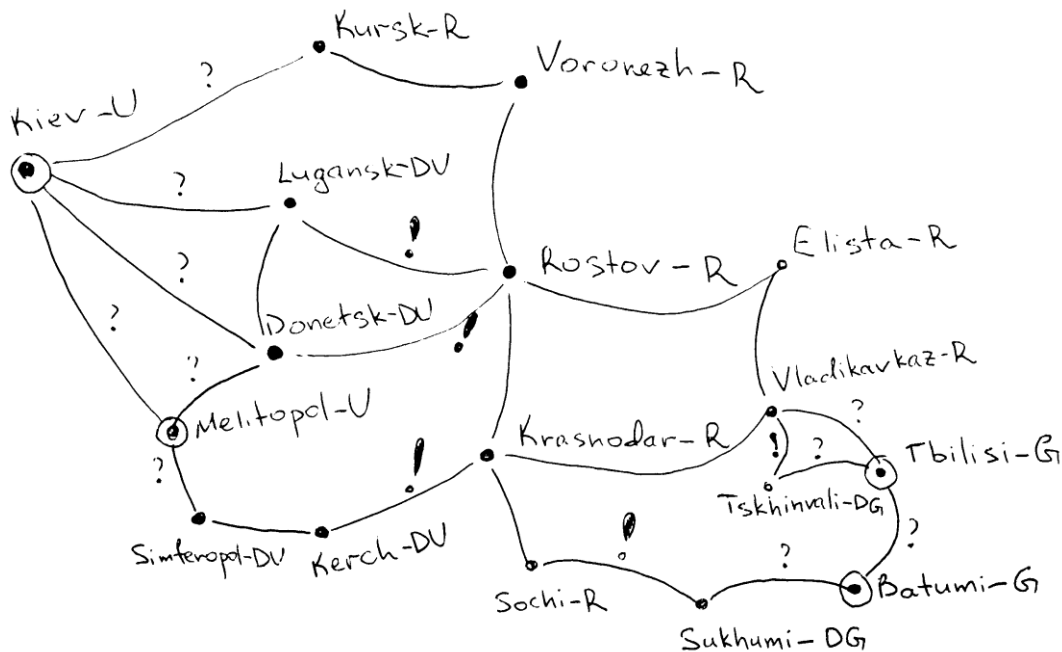


Travelling citizen problems

(Russian version below)

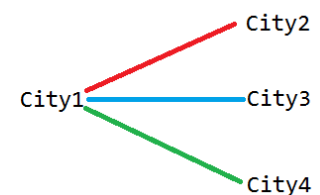
Each problem in this assignment is about the following undirected unweighted graph.



The graph is given as **cities.txt**. Its Format is as follows:

City1 City2 City3 City4

City1 City2 City1 City3 City1 City4



Where the first line lists the vertices, and the second line lists the edges (vertex pairs).

This graph has 2 additional constraints. These constraints are called “violations of rules”:

- 1) When you directly travel from vertex ***-R** to adjacent vertex ***-DU** (marked with "!") or vice versa, all edges from all ***-U** vertices immediately disappear (marked with "?").
- 2) When you directly travel from vertex ***-R** to adjacent vertex ***-DG** (marked with "!") or vice versa, all edges from all ***-G** vertices immediately disappear (marked with "?").

Problem A

Implement **adjacency list graph data structure**; please follow the concepts of lecture slides: you should have **classes for vertices and edges**. Read the graph from the file. Write the method that, for a **given vertex name**, will return **lexicographically ascendingly sorted list of its adjacent vertices**.

- For sorting, use your heap sort implementation.
- Also, include your merge or quick sort implementation into the file – it will be graded.
- Don't take violations of rules into account for this problem.

Example

```
F("Elista-R") = { "Rostov-R", "Vladikavkaz-R" }
F("Vladikavkaz-R") = { "Elista-R", "Krasnodar-R", "Tbilisi-G", "Tskhinvali-DG" }
```

Write result of this method call for the following cities to **around.txt**:

- Donetsk-DU, Kiev-U, Lviv-U, Batumi-G, Rostov-R.

Answer for each city should start from the new line. Separate city names with spaces.

Example:

City list	around.txt
Vladikavkaz-R, Elista-R	Elista-R Krasnodar-R Tbilisi-G Tskhinvali-DG Rostov-R Vladikavkaz-R

For problems B – D, imagine a citizen living in Rostov-R. Any algorithms used for solving these problems must be implemented and then tested for the given input (you cannot solve the task mentally and then just provide the answer).

Problem B

Is this citizen able to visit all cities and get back home without any violations?

Write “yes” or “no” to **able.txt**.

Consider 2 cases:

- 1) Whole initial graph.
- 2) Road (Vladikavkaz-R Tbilisi-G) is closed because of avalanche.

Separate your answers with space.

Problem C

Imagine that 25 years ago there were no “violations of rules”. What was the shortest path for our citizen to:

- 1) Get home from Melitopol-U?
- 2) Travel from Sukhumi-DG to Lugansk-DU?

Write your answers (any valid) to **travel.txt** as follows:

Distance OriginCity CityA CityB ... DestinationCity

Separate your answers for these two cases using a new line.

E.g. given the following example cases “From Kiev-U to Voronezh-R” and “from Batumi-G to Elista-R” the file will contain the following two lines:

```
2 Kiev-U Kursk-R Voronezh-R
3 Batumi-G Tbilisi-G Vladikavkaz-R Elista-R
```

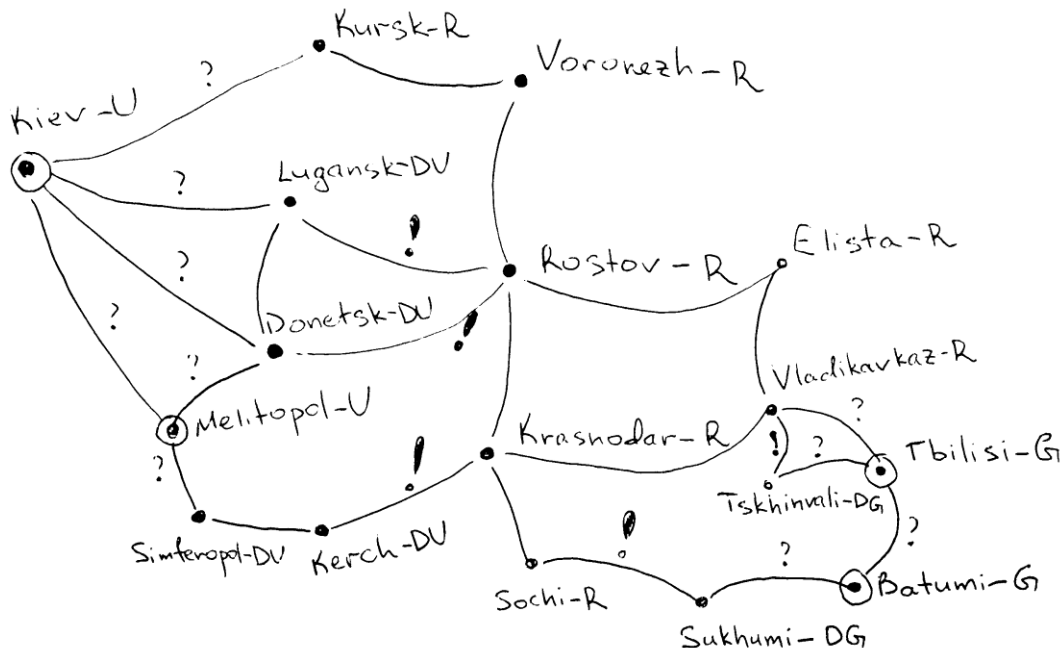
Problem D

Today the rules are in place, and our citizen does not want to violate them.

Answer the same questions (as in C) and write answers to **travel-now.txt**.

Задачи о путешественнике

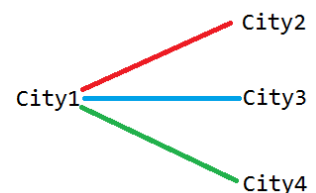
Все задачи данной контрольной работы основаны на приведённом ниже ненаправленном графе без весов.



Граф записан в файле **cities.txt**. Формат файла следующий:

City1 City2 City3 City4

City1 City2 City1 City3 City1 City4



В первой строке приведен список вершин, во второй – список рёбер (пары вершин).

На граф наложены 2 дополнительных ограничения. Назовём их «нарушениями правил»:

- 1) При непосредственном переходе из вершины ***-R** в смежную вершину ***-DU** (такие рёбра отмечены на графе «!») или обратно, все рёбра из

всех *-U вершин немедленно исчезают (такие рёбра отмечены на графе «?»).

- 2) При непосредственном переходе из вершины *-R в смежную вершину *-DG (такие рёбра отмечены на графе «!») или обратно, все рёбра из всех *-G вершин немедленно исчезают (такие рёбра отмечены на графе «?»).

Задача А

Реализуйте структуру данных **список смежности графа (adjacency list)**.

Пожалуйста, ориентируйтесь на концепции, представленные на лекции: у вас должны быть **классы**, для представления **вершин и рёбер**. Считайте граф из файла. Напишите метод, который для заданного имени вершины будет возвращает **отсортированный в алфавитном порядке список смежных вершин**.

- Для сортировки используйте свою реализацию пирамидальной сортировки (heap sort).
- Также, включите в файл свою реализацию быстрой сортировки или сортировки слиянием – она будет оценена.
- Для данной задачи не учитывайте нарушения правил.

Например:

```
F("Elista-R") = { "Rostov-R", "Vladikavkaz-R" }  
F("Vladikavkaz-R") = { "Elista-R", "Krasnodar-R", "Tbilisi-G", "Tskhinvali-DG" }
```

Запишите результаты разработанного метода в файл **around.txt** для следующих городов:

- Donetsk-DU, Kiev-U, Lviv-U, Batumi-G, Rostov-R.

Ответ для каждого очередного города начинается с новой строки. Разделите названия городов пробелами.

Пример:

City list	around.txt
-----------	------------

Vladikavkaz-R, Elista-R	Elista-R Krasnodar-R Tbilisi-G Tskhinvali-DG Rostov-R Vladikavkaz-R
----------------------------	--

|

В задачах В – D мы помогаем найти решения человеку, живущему в Rostov-R. Для решения задач необходимо написать алгоритм и протестировать его на предложенных данных (нельзя просто так взять, придумать и записать ответ).

Задача В

Может ли наш герой посетить все города на карте и вернуться домой без единого нарушения правил? Напишите «yes» или «no» в файл **able.txt**.

Рассмотрите 2 случая:

- 1) Полный исходный граф.
- 2) Дорога (Vladikavkaz-R Tbilisi-G) закрыта из-за схода снежной лавины.

Разделите ответы пробелом.

Задача С

Представьте себе, 25 лет назад не было никаких «нарушений правил». Каким тогда был картчайший путь для нашего героя:

- 1) Чтобы вернуться домой из Melitopol-U?
- 2) Чтобы проехать из Sukhumi-DG в Lugansk-DU?

Напишите свои ответы (любой из корректных) в файл **travel.txt** в следующем формате:

Расстояние ГородОтправления ГородА ГородВ ... ГородНазначения

Разделите ответы для предложенных случаев переносом строки.

Например, для данных примеров – «из Kiev-U в Voronezh-R» и «из Batumi-G в Elista-R» файл будет содержать 2 строки:

```
2 Kiev-U Kursk-R Voronezh-R
3 Batumi-G Tbilisi-G Vladikavkaz-R Elista-R
```

Задача D

Сегодня «правила», которые можно нарушить, в силе. Но наш герой не хочет нарушать их. Для новой вводной ответьте на те же вопросы, что и в предыдущей задаче (C), ответы запишите в **travel-now.txt**.