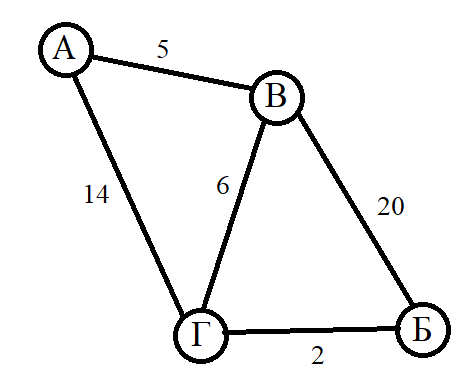
**Лабораторная работа 6 «Маршруты на взвешенном графе»**

Граф, каждому ребру которого поставлено в соответствие некое число (вес) называется взвешенным. Рассмотрим наглядный пример, где вершины – это города, наличие дороги между ними покажем ребром, а вес – длинна соответствующего пути. Путь из в.А в в.Б можно представить, как последовательность рёбер вида (А, к) (к, л) … (м, Б) или последовательность вершин в которой между каждой соседствующей парой есть ребро, например, А, к, л, …м, Б. Будем рассматривать пути без петель, то есть вершины в последовательности не повторяются. Вес маршрута – сумма весов ребер в нем.



На рисунке несколько путей из А в Б вес указан в скобках, АВБ(25), АГБ(16), АВГБ(13).

Для представления взвешенного графа в памяти компьютера используются те же методы, что описаны в Лаб.5 с небольшим дополнением.

**Список ребер –** хранит не только ребро, но и его вес в формате {откуда, куда, вес}, например, {А, Г, 14}

**Матрица смежности –** {i, j}-й элемент соответствует весу ребра из в.i в в.j, 0 - ребро отсутствует.

**Список смежности –** не используется.

Для поиска минимального маршрута рассмотрим алгоритм Дейкстры (<https://ru.wikipedia.org/wiki/Алгоритм_Дейкстры>), который нужно модернизировать для составления самого маршрута в виде списка вершин или ребер.

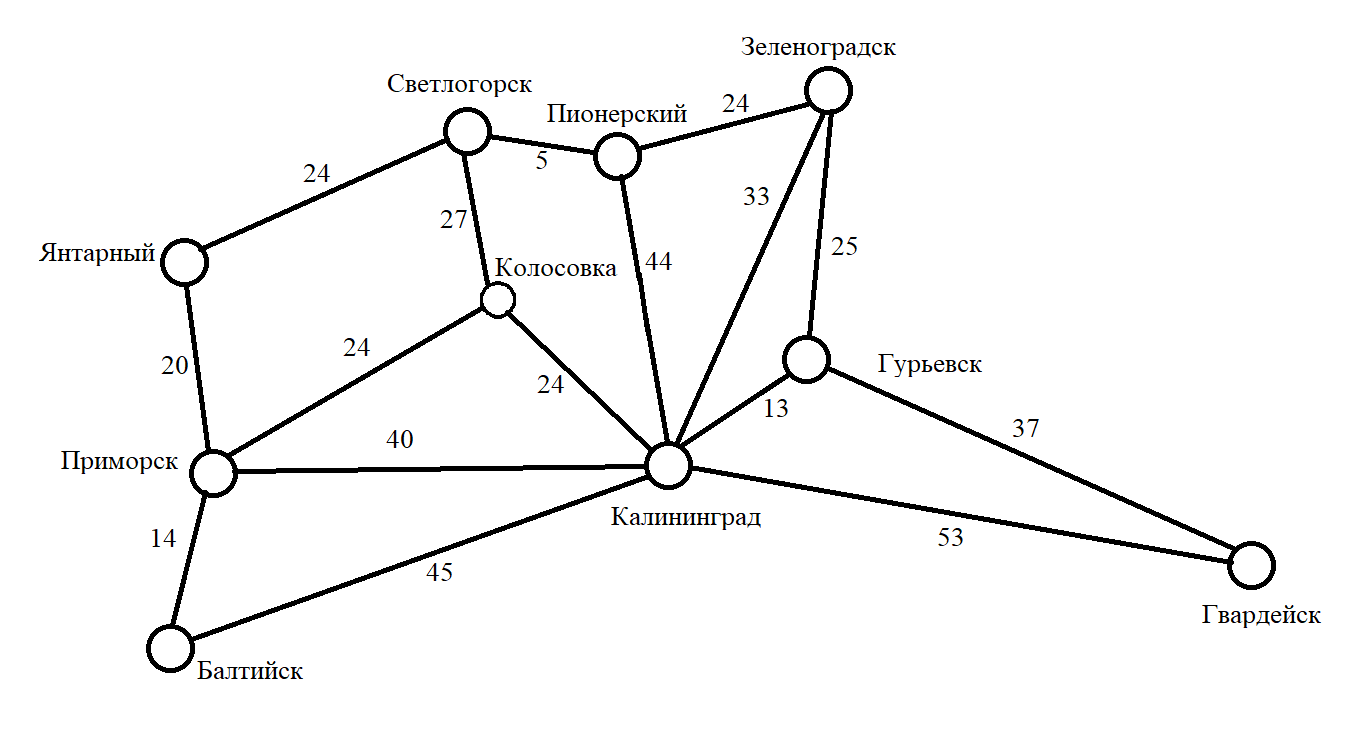
Задание.

1. Вручную преобразовать граф из рисунка в один из форматов.  
2. Написать функцию, которая составляет другое представление графа. (Т.е. если вы преобразовали рисунок в матрицу смежности, то функция по матрице составляет список ребер. А если вы написали список ребер, то функция по этому списку составляет матрицу).  
3. Вывести оба представления в консоль.  
4. Изобразить граф средствами sfml (расположение вершин либо как в лаб5, либо вручную как на картинке). Ребра – прямоугольники серого цвета, вершины – круги красного цвета, вес ребра – по усмотрению.  
5. Написать функцию поиска маршрута из А в Б. (внутренняя функция для п6)  
6. Написать функцию поиска маршрута из А в Б через пункт С. По след. алгоритму:

Составить путь из А в С, изменить граф, чтобы не повторяться (убрать найденный ранее путь), найти путь из С в Б, соединить пути.  
7. Обработать событие «нажатие клавиши пробел» и изобразить граф согласно маршруту из варианта. Вывести маршрут в файл в 2х вариантах (список вершин, список ребер)

Начальная вершина – зеленая, конечная – черная, промежуточные – синие, особая – желтая. Ребра маршрута – светло-зеленые. Пример на картинке ниже.

Карта (для всех)

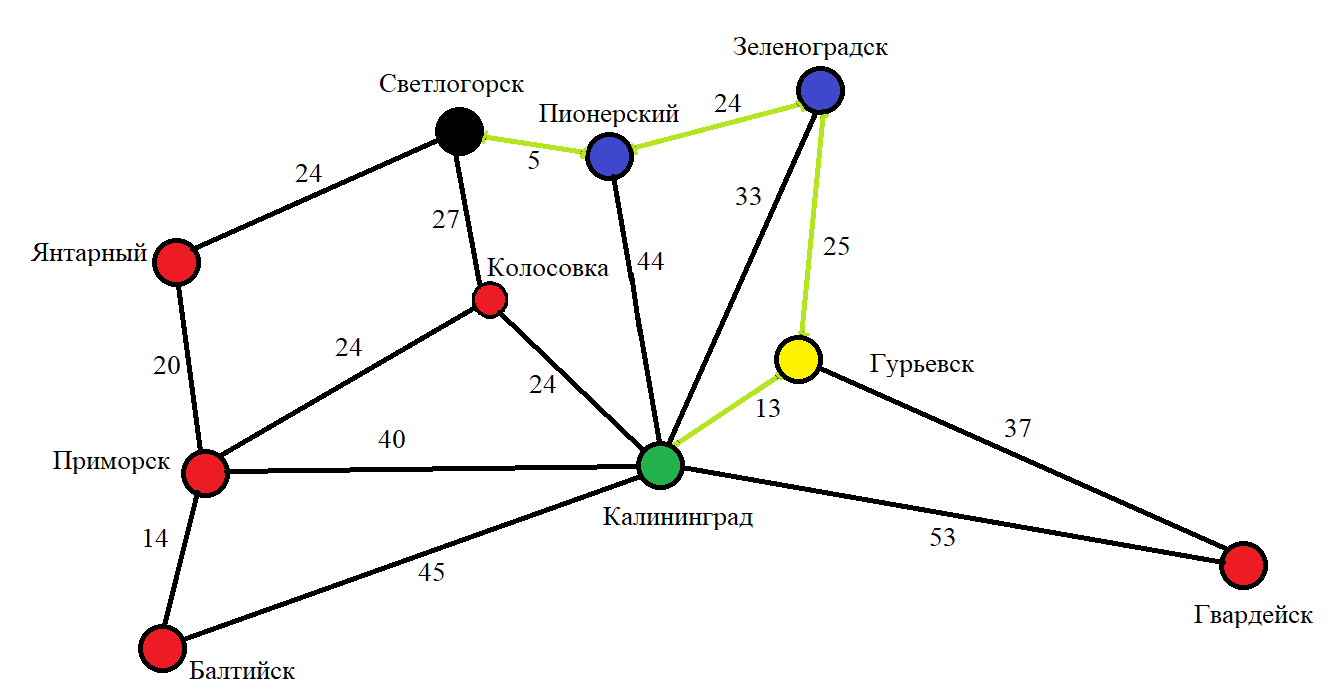


Варианты

|  |  |
| --- | --- |
| 1 | Из Балтийска в Пионерский через Колосовку |
| 2 | Из Балтийска в Гвардейск через Колосовку |
| 3 | Из Зеленоградска в Гвардейск через Калининград |
| 4 | Из Калининграда в Янтарный через Балтийск |
| 5 | Из Светлогорска в Балтийск через Пионерский |
| 6 | Из Гурьевска в Приморск через Колосовку |

Пример

Из Калининграда в Светлогорск через Гурьевск: Калининград-Гурьевск-Зеленоградск-Пионерский-Светлогорск (67км)



**Упрощенная задача. \***

1. Вручную преобразовать граф из рисунка в оба формата и считать их из файла.  
2. Вывести оба представления в консоль.  
3. Изобразить граф средствами sfml (расположение вершин либо как в лаб5, либо вручную как на картинке). Ребра – линии серого цвета, вершины – круги красного цвета, вес ребра – по усмотрению.  
4. Написать функцию поиска маршрута из А в Б.  
5. Обработать событие «нажатие клавиши пробел» и изобразить граф согласно прямому маршруту из варианта, особую точку игнорировать. Вывести маршрут в файл в 2х вариантах (список вершин, список ребер)

|  |
| --- |
| \* в случае возникновения трудностей с выполнением данной задачи вы можете сделать её упрощенный вариант чтобы не потерять баллы допуска. Упрощенная задача лишает возможности автомата, но её задача покрывает допуск к зачету и приносит дополнительные баллы (не превышая максимума). Она не является дополнительным заданием, тк её выполнение – есть часть основной работы. |