Конкурсное задание по логистике

# Постановка задачи

Наша фирма занимается Hi-Tech оборудованием, имеет собственное производство, складские помещения, офисы и торговые точки в общем кол-ве 50 по городу (см приложение 1). Для ритмичной работы всех подразделений мы обязаны регулярно обеспечивать развоз оборудования, используя собственную транспортную службу.

Для экономии средств фирмы мы создаем отдел логистики и ставим перед вами следующую задачу: разработать программу автоматизации выбора оптимального пути передвижения транспорта из произвольного офиса нашей компании (далее точка старта) в несколько подразделений с возвратом в офис в течение одного рабочего дня.

Приложение, по возможности, должно полностью автоматизировать расчет и выбор оптимального пути. Количество объезжаемых подразделений 3< K<10 определяет отдел снабжения. Транспортный отдел внимательно следит за состоянием автотранспорта и охраной труда. Поэтому на планируемый день транспортный отдел задает пороговое значение допустимых суммарных затрат Wpor , которое не должно быть превышено. Необходимо разработать программу с удобным и понятным графическим интерфейсом.

Программа должна находить оптимальный (минимальный) путь объезда максимального количества офисов из заданных отделом снабжения, с суммарными затратами не превышающими Wpor.

В ходе работы программа должна сохранять некоторые промежуточные данные и окончательный результат на диск в соответствии с описанием.

# Исходные данные

В настоящий момент мы составили файл данных содержащий информацию о кратчайших путях между ближайшими подразделениями для всех 50 подразделений компании (см файл *Le-1.txt*). В компании принято передвигаться между подразделениями только по указанному (рекомендованному) маршруту. Поскольку данные задачи предполагают использование теории графов исходные данные подготовлены в виде списка ребер, а именно, в каждой строке первое число - номер подразделения из которого движется транспорт, второе число номер подразделения в который ведет путь, третье число - затраты на транспортировку между этими точками в условных единицах. Расположение офисов на карте показано в файле map.jpg.

Каждая команда получает свой вариант данных, в котором приведены: номер вершины старта, номера офисов от отдела снабжения и пороговое значение затрат от транспортного отдела.

# Формируемые документы

Для обоснования оптимальности выбранного маршрута был принят необходимым следующий перечень документов, которые должны быть сформированы в процессе вычислений и представлены к защите решения.

## Документ 1. Матрица смежности

Матрица смежности, построенная на основе данных текущего списка ребер.

Матрица смежности предъявляется только в электронном виде, в файле с названием *msm.txt*. Данные в файле *msm.txt* при открытии в любом текстовом редакторе должны быть представлены в табличном виде с выравниванием по правому краю с одинаковым шагом между столбцами.

## Документ 2. Список ребер подграфа

Особенность данной транспортной задачи требуют построения рабочего списка оптимальных путей только между заданными подразделениями.

На основании расчетов формируется новый список ребер и, для выполнения контроля корректности хода вычислений, результаты записываются в файл с именем lesg.txt. Представление данных файла полностью соответствует структуре файла le-1.txt:

Данные должны быть представлены в виде таблицы с выравниванием по правому краю с одинаковым шагом между столбцами в следующем формате: первое число номер подразделения из которого движется транспорт, второе число номер подразделения в который ведет путь, третье число - затраты на транспортировку между этими точками в условных единицах, например: 33 4 47.

## Документ 3. Фрагменты кратчайших маршрутов

Одновременно с формированием *lesg.txt* необходимо сформировать массив строк фрагментов маршрута объезда заданных подразделений и сохранить его в файле *pathes.txt* в следующем виде: фрагмент маршрута в одну строку, содержащий список названий начальной, промежуточных и конечной точек в порядке проезда, разделенных набором символов '->' без пробелов.

Например, :

33->50->11->5->23->4

Номер данной строки файла pathes.txt должен совпадать с номером строки "33 4 47" файла Lesg.txt.

Для получения данных для файлов Lesg.txt и Pathes.txt требуется доработка базового алгоритма Дейкстры для нахождения затрат на транспортировку только между заданными подразделениями c детальной трассировкой оптимальных путей.

## Документ 4. Результаты вычислений

В ходе работы программа должна автоматически создавать текстовый файл result.txt следующей структуры (пример) :

11

13 39 17 6 1 24 16

612 650

11-16-39-17-6-1-24-11

11->16->33->39->17->34->12->25->6->1->24->5->11

В первую строку записывается только номер вершины старта. Во вторую строку через пробел номера всех клиентских вершин, заданных отделом снабжения. В третьей строке первое число - полученные минимальные затраты, а второе число - пороговое значение, заданное транспортным отделом. Четвертая строка содержит оптимальную последовательность объезда офисов с возвращением в стартовую вершину. Для получения затрат меньше пороговых офис с номером 13 пришлось оставить на "другой день". Номера офисов в четвертной строке разделены символом "-" (знак минус) без пробелов. Последняя пятая строка должна содержать подробный маршрут движения автотранспорта по заданным ребрам графа. Номера вершин разделяются символами "минус" и "больше" без пробелов.

## Документ 5. Лист утверждения маршрута (ЛУМ)

Данный документ формата word - файл "лум.docx" - создается на базе шаблона "лум\_шаблон.docx", который хранится в общей папке с файлами "map.jpg" и "Le-1.txt". Пример заполнения документа лум.docx можно посмотреть в файле "лум\_пример.docx".

## Требования к приложению

## Интерфейс приложения

Интерфейс приложения должен быть выполнен в едином стиле, содержать иконку предприятия и не иметь нефункциональных элементов управления.

Интерфейс должен включать в себя следующие элементы управления:

1. Поле ввода номера офиса старта
2. Поле ввода номеров посещаемых филиалов
3. Кнопку запуска вычислений
4. Текстовое поле для вывода результата вычислений
5. Элементы управления должны быть корректно подписаны.

## Пользовательская документация

Инструкция пользователя должна описывать функционал приложения, методику использования и решение возможных нештатных ситуаций.

Вся представленная документация не должна содержать смысловых и орфографических ошибок.

Полученный маршрут объезда должен быть отображен на электронном варианте карты и представлен при защите решения.

## Исходный код приложения

1. Исходный код должен быть оформлен, используя общепринятые стандарты оформления кода на используемом языке программирования;
2. Используемые методы должны быть документированы.
3. Компилятор не должен выдавать предупреждений о неиспользуемых переменных.

## Требования к надежности

1. Стабильная работа приложения при штатном использовании;
2. Приложение должно корректно обрабатывать любые случайные ошибки пользователя, например неполный или некорректный ввод данных.