

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

НОВГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ЯРОСЛАВА МУДРОГО

НАХОЖДЕНИЕ КРАТЧАЙШЕГО ПУТИ И ВЫБОР ОПТИМАЛЬНОЙ СХЕМЫ АВТОБУСНЫХ МАРШРУТОВ С ПОМОЩЬЮ ПЭВМ

Методические указания

ВЕЛИКИЙ НОВГОРОД
2017

УДК 629.113.004
НЗ4

Печатается по решению
РИС НовГУ

Р е ц е н з е н т ы:

кандидат технических наук, доцент **И. И. Зубрицкас**
кандидат технических наук, доцент **А. Н. Чадин**

Нахождение кратчайшего пути и выбор оптимальной схемы
НЗ4 автобусных маршрутов с помощью ПЭВМ: метод. указания /
авт.-сост. С. В. Гудилов; НовГУ им. Ярослава Мудрого. – Великий
Новгород, 2017. – 60 с.

В методических указаниях рассмотрены вопросы нахождения экстремумов функций одной и нескольких переменных с помощью ПЭВМ. Приводятся примеры и программы расчета.

Предназначены для студентов направления 23.03.03. «Эксплуатация транспортно-технологических машин и комплексов» для практических занятий по курсу «Основы инженерных расчетов».

УДК 629.113.004

© Новгородский государственный
университет, 2017

© С. В. Гудилов, составление, 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1. ЦЕЛЬ РАБОТЫ	5
2. ЗАДАЧА НАХОЖДЕНИЯ КРАТЧАЙШЕГО ПУТИ	5
2.1. Моделирование транспортных сетей	5
2.2. Определение кратчайшего пути на графе методом динамического программирования.	6
3. ВЫБОР СХЕМЫ АВТОБУСНЫХ МАРШРУТОВ В ГОРОДАХ.....	7
3.1. Постановка задачи.....	7
3.2. Метод решения	8
3.3. Постановка задачи на ПЭВМ.....	13
4. ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ РАБОТЫ	15
5. СОДЕРЖАНИЕ ОТЧЁТА.....	15
6. ТЕХНИКА БЕЗОПАСНОСТИ.....	15
СПИСОК ЛИТЕРАТУРЫ.....	15
<i>Приложение 1</i>	16
<i>Приложение 2</i>	19
<i>Приложение 3</i>	26
<i>Приложение 4</i>	38

ВВЕДЕНИЕ

В настоящее время с целью дальнейшего совершенствования планирования на автомобильном транспорте значительное внимание стало уделяться широкому применению в планировании экономико-математических методов. Расчеты с помощью экономико-математических методов обычно требуют большого объема вычислений, и для их проведения используется ПЭВМ, что обеспечивает возможность из многих вариантов решения выбрать наилучший, оптимальный вариант.

В математической форме задача оптимального планирования формулируется следующим образом. Существует система величин, которые могут принимать различные значения в заданных пределах. Требуется найти такие значения этих величин, которые оптимизируют выбранный критерий, являющийся их функцией.

Среди математических методов решения планово-экономических задач прежде всего следует назвать методы математического программирования, с помощью которых решаются задачи на экстремум функции многих переменных с ограничениями на область изменения этих переменных. Из методов математического программирования при решении планово-экономических задач наибольшее распространение получили методы линейного программирования. Слово «линейное» определяет математическую природу моделей, с помощью которых описываются эти задачи. Она состоит в том, что условия задач выражаются системой линейных уравнений или неравенств, содержащих неизвестные только первой степени.

Для решения планово-экономических задач, связанных с принятием ряда последовательных и поэтапных решений, используются методы динамического программирования, в основе которых лежит совокупность приемов, позволявших находить оптимальные решения, основанные на вычислении последствий каждого из них и выработке оптимальных стратегий для последующих решений.

Кроме методов математического программирования в решении планово-экономических задач находят применение и методы, которые созданы в других прикладных направлениях математической науки. Примером этого могут служить методы комбинаторного анализа, которые используются при составлении развозочных и сборных маршрутов перевозок мелкопартионных грузов и при выборе схем автобусных маршрутов в городах.

1. ЦЕЛЬ РАБОТЫ

Научиться моделировать кратчайшие пути на графе методом динамического программирования и выбирать оптимальные схемы автобусных маршрутов в городах.

2. ЗАДАЧА НАХОЖДЕНИЯ КРАТЧАЙШЕГО ПУТИ

2.1. Моделирование транспортных сетей

При планировании перевозок возникает необходимость в определении кратчайших расстояний между АТП, пунктами производства и потребления. Кроме того, кратчайшие расстояния являются основой при оплате клиентами транспортных услуг. Они необходимы также для определения грузооборота АТП, учета расхода топлива, расчета заработной платы водителей и т.п. Для нахождения оптимального решения в данной задаче в качестве исходных данных необходима транспортная сеть, отражающая транспортные связи между точками города (местности). Модель такой сети может быть представлена в виде графа.

Граф – это фигура, состоящая из точек (вершин) и отрезков (ребер), их соединяющих (рис. 2.1). Ребра характеризуются числами, которые могут иметь различный физический смысл.

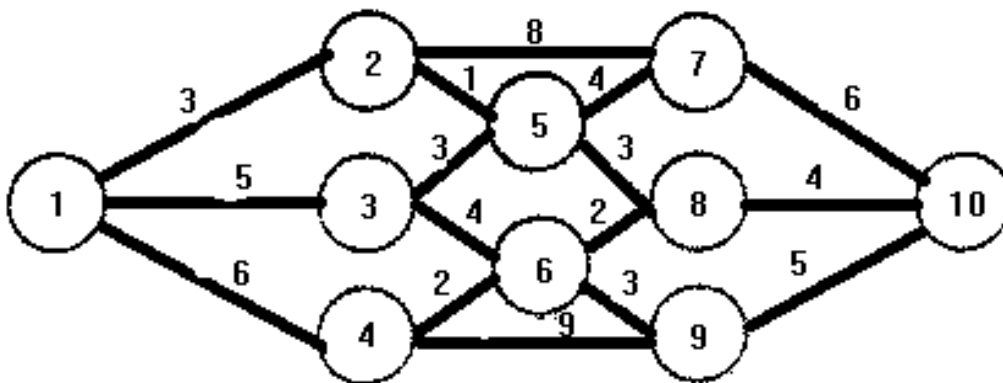


Рис. 2.1. Граф транспортной сети

Ребра, ориентированные по направлению, называются *дугами*. В зависимости от того, все или часть ребер имеют направление, граф является *ориентированным* или *смешанным*. Граф, каждая вершина которого может быть соединена некоторой последовательностью ребер с любой другой его вершиной, называется *связным графом*.

Граф, моделирующий транспортную сеть, **обязательно должен быть связным**, чтобы всегда был путь из любой вершины в любую дру-

гую. Числа, характеризующие ребра такого графа, выражают протяженность пути, время или стоимость проезда. Моделирование транспортной сети начинают с размещения вершин. Вершины присваивают точкам города, соответствующим основным грузообразующим и грузопоглащающим пунктам, центрам крупных жилых кварталов, пересечениям улиц.

В условиях крупного города при оперативном планировании перевозок необходимо быстро определить расстояния между большим числом поставщиков, потребителей и автохозяйств. Если вести расчеты на ПЭВМ, то для хранения таблиц расстояний потребуется слишком много памяти. Поэтому для сокращения размеров таблиц карту города разбивают на микрорайоны. Построив для микрорайона таблицу кратчайших расстояний, легко определить расстояние между любыми точками города.

2.2. Определение кратчайшего пути на графе методом динамического программирования

Существует целый ряд различных математических методов поиска кратчайшего пути. Наиболее эффективным для поиска кратчайшего пути на графе является *метод динамического программирования* [2].

В общем виде метод динамического программирования можно определить как метод оптимизации многошаговых процессов принятия решений. Метод заключается в том, что каждый i -й шаг оптимизируется не отдельно, а с учетом его последствий в перспективе, т.е. действия на весь последующий процесс, начиная с i -го шага. Причем принимается, что состояние процесса, достигнутое на предыдущих $(i-1)$ шагах уже оптимизировано и учитывается на i -м шаге. Из этого следует, что необходимо найти такой шаг, который не влиял бы на продолжение процесса, и в то же время его можно было бы планировать как оптимальный. Таким шагом является *окончание процесса*. Поэтому при оптимизации процесса методом динамического программирования рассмотрение его начинается с конца процесса, что является особенностью этого метода.

Метод обладает следующими достоинствами:

- 1) упрощает поиск решения вариационных задач, а также резко сокращает объем вычислений в задачах с большим числом вариантов;
- 2) метод допускает широкое применение вычислительной техники, что облегчает решение сложных задач.

Однако метод обладает и одним существенным недостатком — отсутствием универсального алгоритма, который был бы пригоден для решения всех задач. Используемые в настоящее время алгоритмы динамического программирования объединены лишь общей идеей и в каждом конкретном случае разрабатываются применительно к условиям конкретной задачи.

В прил. 1, 3 приводятся программы нахождения кратчайшего пути на графе методом динамического программирования, позволяющие эффек-

тивно решать задачу с числом вершин на графе до $N \leq 100$. Для расчета кратчайшего пути используется рекуррентное выражение:

$$f_j = \min(f_i + i_{ij}), \quad j = \overline{1, N},$$

где f_j – длина кратчайшего пути, соединяющего начальную и j -ю вершины;
 i_{ij} – длина дуги, соединяющей вершину i с вершиной j .

3. ВЫБОР СХЕМЫ АВТОБУСНЫХ МАРШРУТОВ В ГОРОДАХ

3.1. Постановка задачи

В основе решения задачи по выбору схемы автобусных маршрутов в городах лежат математические методы комбинаторного анализа. В данном случае используется предложенный В.А. Паршиковым метод с направленным отбором вариантов, который позволяет находить оптимальное решение в 95–98 случаях из 100 [1].

Имеется транспортная сеть – улицы города, по которым возможно движение автобусов. Заданы крупные пункты зарождения и погашения пассажиропотоков – узлы и соединяющие их участки улиц – звенья транспортной системы. Установлены размеры пассажиропотоков между узлами заданной транспортной сети и типы автобусов, обслуживающих намечаемые линии, и их характеристики.

Требуется определить такую схему автобусных маршрутов, чтобы суммарные затраты времени пассажирами на ожидание, проезд и пересадки были минимальными. При этом на решение могут быть наложены следующие ограничения: использование вместимости автобусов должно быть не ниже заданного коэффициента, интервал между отправлениями автобусов не может превышать заданной величины, различной для разных линий, протяженность маршрута должна быть не меньше минимальной и не больше максимальной длины, которая задается заранее, маршруты не должны начинаться и заканчиваться в тех узлах, которые не могут быть использованы для организации конечных пунктов маршрутов, и другие ограничения, вытекающие из местных условий каждого конкретного города.

Исходя из указанной формулировки задачи выбора схемы автобусных маршрутов в городах, для ее решения необходимы следующие основные исходные данные:

1. Карта города с транспортной сетью, состоящей из пунктов зарождения и погашения пассажиропотоков и улиц, соединяющих эти пункты, по которым возможно движение автобусов.

Под пунктами зарождения и погашения пассажиропотоков обычно понимают *транспортные микрорайоны города*. При разбивке города на

микрорайоны в первую очередь используются естественные и искусственные рубежи (реки, железнодорожные линии и т.д.). Транспортные магистрали при этом по возможности должны быть осями симметрии микрорайона, площадь которого в пределах 250–350 га, что обеспечивает подход пассажиров к остановочным пунктам не более 700 метров. Поэтому при решении данной задачи принимается, что пешие переходы до и от остановки не зависят от схемы маршрутов и в расчетах не учитываются. На транспортной сети указываются длина каждого ее участка и время следования автобуса по этим участкам.

2. Размеры пассажиропотоков между всеми пунктами города, которые определяются на основе анкетного обследования пассажиропотоков, при этом в каждой анкете указывается – откуда и куда следует пассажир, что позволяет при обработке анкет определить соответствующие микрорайоны начала и окончания поездок пассажиров. Наиболее целесообразно маршрутную схему разрабатывать на основе трудовых и других поездок в утренние часы пик в зимнее время.

3. Используемая вместимость единицы подвижного состава с учетом заданного коэффициента заполнения, обеспечивавшего предоставление пассажирам необходимых удобств поездки.

4. Время, затрачиваемое одним пассажиром на пересадки в каждом пункте.

5. Максимальные (в некоторых случаях минимальные) интервалы движения автобусов.

6. Минимальный коэффициент использования вместимости автобусов по всей сети маршрутов в целом, обеспечивающий определенное эффективное использование имеющегося или планируемого парка автобусов.

3.2. Метод решения

Методику расчета схемы автобусных маршрутов в городе рассмотрим на условном, несколько упрощенном примере. Упрощение состоит в том, что в примере не учитываются все возможные ограничения, которые могут быть в реальных условиях того или иного города, однако основные из них в данном примере заданы [1].

Транспортная сеть рассматриваемого примера представлена на рис. 3.1. Цифры в кружках означают номера пунктов; цифры в середине участков цепи указывают время движения автобусов по данному участку в минутах, а цифры в скобках – длину участка в километрах.

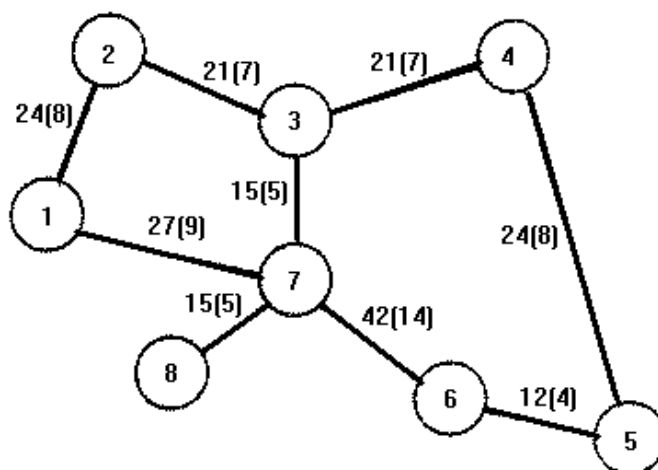


Рис. 3.1. Транспортная сеть

Пассажиропоток в расчетный период в час пик, принятый в данном примере за один час, представлен в табл. 3.1.

Эта таблица составляется на основе обработки результатов обследования пассажиропотоков.

Все маршруты обслуживаются одним типом автобусов, вместимость каждого из которых с учетом коэффициента наполнения составляет 40 человек. Максимальный интервал между отправлениями автобусов $I_{max} = 12$ минут. Время движения между пунктами в обе стороны принято равным, хотя в общем случае оно может быть различным.

Таблица 3.1

Корреспонденция пассажиропотоков в час пик

Пункты отправления	Пассажиропоток, чел.							
	Пункты прибытия							
	1	2	3	4	5	6	7	8
1	—	200	60	70	56	120	70	210
2	220	—	300	350	20	130	250	190
3	60	56	—	240	540	105	30	110
4	20	350	80	—	60	10	360	30
5	70	40	600	46	—	150	80	20
6	45	85	90	50	60	—	100	40
7	180	235	100	520	90	250	—	85
8	335	185	25	64	40	100	70	—

Время, затрачиваемое одним пассажиром на пересадку в каждом пункте, представлено в табл. 3.2.

Разработка маршрутной схемы состоит из нескольких этапов.

Этап 1. Определение кратчайших (по времени) путей между пунктами (микрорайонами)

Таблица 3.2

Время, затрачиваемое пассажирами на пересадки

№ пункта	1	2	3	4	5	6	7	8
Время, мин	2	3	3	5	4	4	5	3

Алгоритм решения состоит из 2-х шагов.

Шаг 1. Присвоим начальному узлу сети потенциал 0.

Шаг 2. Просмотрим все звенья, начальные узлы которых имеют потенциалы, а конечные не имеют. Определим потенциалы конечных узлов как сумму потенциала начального узла и времени движения автобуса по звену, соединяющему начальный и конечный узлы. Выбираем конечный узел с наименьшим потенциалом, запишем его рядом с узлом и отметим звено стрелкой. Шаг 2 повторяем до тех пор, пока всем узлам не будут присвоены потенциалы.

В нашем примере первым начальным узлом примем пункт 1. Присвоим ему потенциал 0 (рис. 3.2). На шаге 2 надо сопоставить 2 потенциала: $0+24$ и $0+27$ и выбрать конечный пункт с минимальным потенциалом. Это пункт 2.

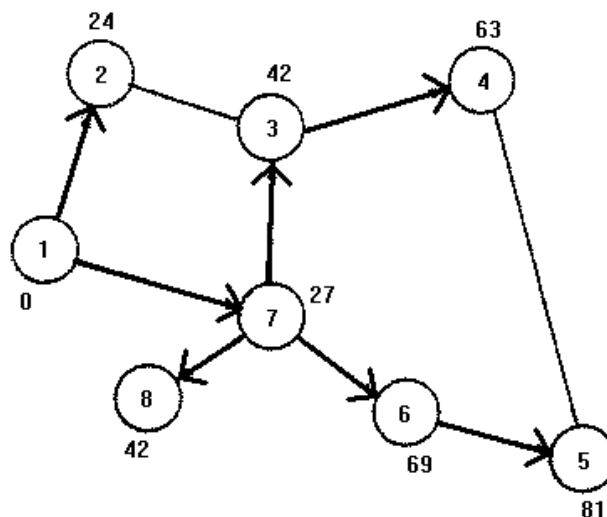


Рис. 3.2. Определение кратчайших путей методом потенциалов

Записываем около этого пункта потенциал 24, а звено 1–2 отмечаем стрелкой. Повторяем шаг 2. Сравниваем результаты: $0+27 = 27$, $24+21 = 45$. Выбираем наименьший из этих потенциалов, который относится к пункту 7. Записываем его около этого пункта, а звено 1–7 помечаем стрелкой. Продолжая этот процесс, получаем потенциалы всех пунктов. Звенья со стрелочкой показывают кратчайший путь от пункта 1 ко всем остальным пунк-

там. Результаты этих расчетов записываем в табл. 3.3, где в соответствующих клетках в верхнем левом углу указаны пункты, через которые проходит кратчайший путь, а внизу – время следования между начальным и конечным пунктами. Из табл. 3.3 видно, что кратчайший путь из пункта 1 в пункт 4 проходит через пункты 3 и 7 и время следования (без учета времени на пересадки) составит 63 минуты.

Аналогично выполняются расчеты по всем пунктам.

Таблица 3.3

Кратчайшие пути маршрутов и время движения по ним

Пункты отправления	Время движения, мин							
	Пункты прибытия							
	1	2	3	4	5	6	7	8
1	–	24	7 42	7; 3 63	7; 6 81	7 69	27	7 42
2	24	–	21	3 42	3; 7; 6 89	3; 7 77	3 35	3; 7 50
3	7 42	21	–	21	7; 6 69	7 57	15	7 30
4	3; 7 63	3 42	21	–	84	3; 7 78	3 35	3; 7 51
5	6; 7 81	6; 7; 3 89	6; 7 69	84	–	12	6 54	6; 7 69
6	7 69	7; 3 77	7 57	7; 3 78	12	–	42	7 57
7	27	3 35	15	3 36	6 54	42	–	15
8	7 42	7; 3 50	7 30	7; 3 51	7; 6 69	7 57	15	–

Этап 2. Установление исходной схемы

В качестве исходной маршрутной схемы принимается схема, в которую входят маршруты, удовлетворяющие достаточному условию назначения беспересадочных сквозных маршрутов, а также участковые маршруты, не совпадающие ни с одним маршрутом. В качестве сквозного рассматривается маршрут, соединяющий центры 3-х и более микрорайонов по кратчайшему пути, исходя из затрат времени на следование в пути. Достаточным условием для назначения сквозного маршрута является удовлетворение естественного требования, чтобы время ожидания пассажиром автобу-

са на начальном пункте маршрута было бы меньше или равно времени, которое он должен затратить в пункте пересадки, если того маршрута не будет, т.е. будет выдержано следующее соотношение:

$$\frac{CgT_p}{\rho} \frac{1}{P_{ij}} \leq t_{nl},$$

где C – коэффициент равномерности подхода пассажиров к остановке (принимается равным 0,5);

g – используемая вместимость автобуса, чел.; $g = 40$ чел.;

T_p – продолжительность расчетного периода суток, мин; $T_p = 60$ мин;

ρ – коэффициент внутри часовой неравномерности пассажирского потока; $\rho = 1,1$;

P_{ij} – число пассажиров, проезжающих между конечными пунктами назначаемого маршрута в направлении максимального пассажиропотока, чел.;

t_{nl} – затраты времени на пересадку одного пассажира в пункте, имеющем минимальную продолжительность пересадки по сравнению с другими промежуточными пунктами по пути между начальным i и конечным j пунктами назначаемого сквозного маршрута в направлении максимального пассажиропотока, мин.

Маршруты, отвечающие этому условию, включаются в исходный вариант схемы автобусных маршрутов.

Таким образом проверяются все маршруты в рассматриваемом примере. По условию в исходную схему маршрутов будут назначены маршруты 1–8, 3–5, 4–7.

С ними не совпадают участковые маршруты 1–2, 2–3 и 4–5, которые также должны быть включены в исходный вариант схемы автобусных маршрутов. Таким образом, исходная схема автобусных маршрутов состоит из шести маршрутов: 1–8, 3–5, 4–7, 2–3, 1–2 и 4–5.

Исходная схема представлена на рис. 3.3.

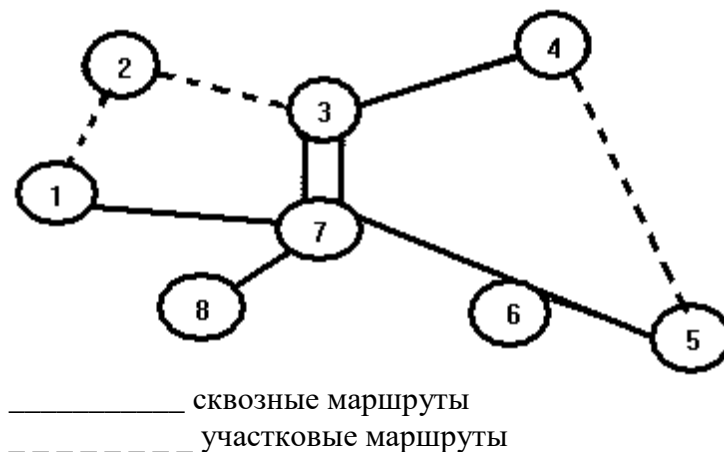


Рис. 3.3. Исходная схема

Этап 3. Проверка участковых маршрутов на соответствие заданному интервалу движения

Проверке подлежат только те участковые маршруты, которые не совпадают со сквозными и проходят через пункты, между которыми есть возможность проезда на автобусах по другим маршрутам.

В рассматриваемом примере таким участковым маршрутом является маршрут 4–5. Для расчета интервала движения автобуса по маршруту принимается направление с наибольшим пассажиропотоком P_{ij}^{max} .

Интервал движения определяется по формуле

$$I_{ij} = \frac{gT_p}{P_{ij}}.$$

Для маршрута 4–5 этот интервал равен $I_{4-5} = \frac{40 \cdot 60}{60} = 40$ мин.

Так как задаваемый максимальный интервал движения равен 12 минутам, а по расчету получен интервал 40 минут, то этот маршрут нужно исключить из исходного варианта маршрутной схемы как не отвечающий заданному ограничению. Пассажиры из пункта 4 в пункт 5 и обратно будут ездить через пункты 3, 7, 6.

Пункт 2 с остальной транспортной сетью связывают только участковые маршруты 1–2 и 2–3. Проверим их соответствие заданному максимальному интервалу движения.

Маршрут 1–2: $I_{1-2} = \frac{40 \cdot 60}{220} = 11$ мин.

Маршрут 2–3: $I_{2-3} = \frac{40 \cdot 60}{300} = 8$ мин.

Оба маршрута должны быть оставлены в исходном варианте схемы маршрутов, так как интервал движения по ним ниже заданного максимального. Однако заметим, что если оба этих маршрута не удовлетворяли бы этому ограничению, то в исходном варианте все равно должен был бы быть оставлен один из них (тот, у которого был бы минимальный интервал движения), так как связь пункта 2 с остальными пунктами должна быть обеспечена.

Таким образом, исходным вариантом маршрутной схемы в данном примере является вариант с пятью маршрутами: 1–8, 3–5, 4–7, 1–2 и 2–3.

3.3. Постановка задачи на ПЭВМ

В прил. 2 представлена программа выбора схемы автобусных маршрутов в городе.

В строках 590–1560 определяются кратчайшие пути между пунктами транспортной сети и устанавливается исходная маршрутная схема. Переменная H – начало маршрута, K – конец. Для того чтобы не было прямых и обратных маршрутов (т.е. 1–2 и 2–1), принимаем всегда $K > H$ (строка 610). Тем самым мы отсекаем половину всех возможных вариантов. После выбора начального и конечного пунктов программа шагает от начального пункта по связям $C(I, J)$ до тех пор, пока не дойдет до конечного пункта. При этом программа идет по всем возможным вариантам V . Шаг R между пунктами, при этом в операторе $N(V, R)$ запоминаются все пройденные маршрутом пункты. Переменная $V5$ показывает текущее значение вариантов пути. В операторе $L3(V, R)$ запоминается текущее значение длины пути. После того как один из путей достигает пункта K , этот путь принимается за основной. Его шаги запоминаются в операторе $N2(R)$, длина маршрута – в операторе $L0$.

Теперь продолжают только те пути, длина которых меньше длины основного маршрута $L0$, на это указывает индикатор $L1(V)$. Пока длина пути меньше $L0$, его значение I , при невыполнении условия его значение меняется на 0, и в дальнейшем этот путь не рассматривается (строка 760). Если путь все-таки достигает пункта K , то он принимается за основной. После того как найден оптимальный вариант, ему присваивается номер M и запоминается начало маршрута $H4(M)$, конец – $K4(M)$, длина – $L4(M)$ и шаги в операторе $N4(M, R)$. После этого, если маршрут содержит более 3 пунктов, он проверяется на условие, что время пересадки $T1$ будет больше времени ожидания $T0$. $T1$ вычисляется в строках 1420–1450. $P8$ – наибольшее из значений пассажиропотоков $P(H, K)$ и $P(K, H)$; $C0$ – коэффициент неравномерности подхода пассажиров к остановке; $Q0$ – используемая вместимость автобуса; $T8$ – продолжительность расчетного периода суток; $P0$ – коэффициент внутрисуточной неравномерности пассажирского потока. Если маршрут проходит по условию, то его номер запоминается в операторе $M(F)$.

В строках 1570–1610 идет дублирование массива связей $C(I, J)$ в массив $C1(I, J)$. Затем в строках 1620–1860 идет удаление из массива $C1(I, J)$ тех связей, которые были задействованы при выборе исходного варианта. Вместо пункта связи в массиве $C1(I, J)$ ставится 1. После выполнения этой процедуры в новом массиве определяются неиспользованные участковые маршруты и проверяется интервал движения на нем SI и сверяется с максимальным по условию $S0$. Интервал определяет пассажиропоток на участке.

Если условие выполняется, то этот участковый маршрут вносится в основную схему маршрутов, об этом свидетельствует индикатор $W(M)$. Далее проверяется – все ли пункты соединены между собой исходной схемой маршрутов; если нет, то в строках 2110–2640 проверяются на интервал все участковые маршруты и их значения (запоминаются в операторе $S4(B)$), связывающие этот пункт с остальными, охваченными исходной

схемой маршрутов. Затем выбирается маршрут с наименьшим интервалом и вносится в исходную схему. После этого на экран выводятся данные обо всех маршрутах исходной сети (строки 2660–2770).

4. ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ РАБОТЫ

1. Ознакомиться с теоретическими положениями (п. 2, 3).
2. Ознакомиться с работой программ (прил. 1, 2, 3, 4) на контрольных примерах.
3. Получить у преподавателя индивидуальное задание.
4. Внести необходимые изменения в программу прил. 2 или прил. 4, выполнить задание.

5. СОДЕРЖАНИЕ ОТЧЁТА

1. Текст исходного задания.
2. Распечатка полученных результатов.

6. ТЕХНИКА БЕЗОПАСНОСТИ

1. При выполнении лабораторной работы следует соблюдать требования техники безопасности, необходимые при работе с электрическими приборами с напряжением до 220 В.
2. В случае обнаружения неисправности ПЭВМ необходимо сообщить об этом преподавателю.

СПИСОК ЛИТЕРАТУРЫ

1. *Белолипецкий А. А.* Экономико-математические методы : учеб. для вузов по спец. "Экономика" / А. А. Белолипецкий, В. А. Горелик. - М. : Академия, 2010. – 362.
2. *Геронимус, Б. Л.* Экономико-математические методы в планировании на автомобильном транспорте / Б. Л. Геронимус, Л. В. Царфин. М.: Транспорт, 1988. 192 с.
3. *Дягтерев, Ю. И.* Исследования операций / Ю. И. Дягтерев. М.: Высш. шк., 1986. 320 с.

```
10 CLS
15 COLOR 11
20 PRINT "ПРОГРАММА ОПРЕДЕЛЕНИЯ КРАТЧАЙШЕГО ПУТИ"
30 PRINT " В ОРИЕНТИРОВАННОМ ГРАФЕ МЕТОДОМ"
40 PRINT " ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ"
50 FOR I = 1 TO 38
60 PRINT "*";
70 NEXT I
80 PRINT
90 K = 0
95 COLOR 12
100 PRINT "ВВЕДИТЕ КОЛИЧЕСТВО ВЕРШИН ГРАФА"
110 INPUT N
120 DIM P(N, N)
125 COLOR 13
130 FOR I = 1 TO N - 1
140 PRINT "ВЕРШИНА"; I; "СВЯЗАНА С ВЕРШИНОЙ";
150 INPUT L
160 IF L = 0 THEN 210
170 PRINT "ДЛИНА ДУГИ"; I; "-"; L; "ГРАФА";
180 INPUT P(I, L)
190 P(L, I) = P(I, L)
200 GOTO 140
210 NEXT I
220 GOSUB 740
230 FOR W = N - 1 TO 1 STEP -1
240 I = W + 1
250 GOSUB 450
260 IF K = 0 THEN K = N
270 FOR I = W TO N
280 IF I = K THEN 300
290 P(I, W) = 0
300 NEXT I
310 P(W, N) = S
320 PRINT "W="; W; "S="; S
330 GOSUB 740
340 NEXT W
350 J = 1
360 PRINT
370 FOR I = 1 TO 33
380 PRINT "*";
390 NEXT I
```


Продолжение приложения 1

```
400 PRINT
405 COLOR 11
410 PRINT "ОПТИМАЛЬНЫЙ ВАРИАНТ ПУТИ В ГРАФЕ:"
420 PRINT "1";
430 GOTO 560
450 IF P(I, W) = 0 THEN 810
460 IF I = N THEN 880
470 S = P(I, W) + P(I, N)
480 K = I
490 I = I + 1
500 IF P(I, W) = 0 THEN 860
510 S1 = P(I, W) + P(I, N)
520 IF S1 < S THEN 830
530 I = I + 1
540 IF I <= N THEN 500
550 RETURN
560 FOR I = J + 1 TO N
570 IF P(I, J) <> 0 THEN 590
580 NEXT I
590 J = I
600 PRINT "-";
610 PRINT J;
620 IF J <> N THEN 560
630 PRINT
635 COLOR 10
640 FOR I = 1 TO 28
650 PRINT "*";
660 NEXT I
670 PRINT
680 PRINT "ДЛИНА ОПТИМАЛЬНОГО ПУТИ T="; P(1, N)
690 FOR I = 1 TO 28
700 PRINT "*";
710 NEXT I
720 PRINT
730 END
740 FOR I = 1 TO N
750 FOR J = 1 TO N
760 PRINT P(I, J);
770 NEXT J
780 PRINT
790 NEXT I
795 INPUT ZZ
```

Окончание приложения 1

```
800 RETURN
810 I = I + 1
820 GOTO 450
830 S = S1
840 K = I
850 GOTO 530
860 I = I + 1
870 GOTO 540
880 S = P(I, W)
890 RETURN
```

```
10 CLS
15 COLOR 10
20 PRINT TAB(17); "ПРОГРАММА ОПТИМИЗАЦИИ СЕТИ"
30 PRINT TAB(15); "ГОРОДСКИХ АВТОБУСНЫХ МАРШРУТОВ"
40 DIM C(8, 6), T(8), P(8, 8), L(8, 8), N(30, 8), N4(30, 8)
50 DIM H4(30), K4(30), L4(30)
60 DIM N2(10), L1(20), L3(30, 8), M(30), C1(8, 6), S4(10), W(30)
70 Q = 6
80 N = 8
90 C0 = .5
100 Q0 = 40
110 S0 = 12
120 T8 = 60
130 P0 = 1.1
140 DATA 1,2,3,4,5,6,7,8
150 DATA 2,3,2,3,4,5,1,7
160 DATA 7,1,4,5,6,7,3,0
170 DATA 0,0,7,0,0,0,6,0
180 DATA 0,0,0,0,0,0,8,0
190 DATA 0,0,0,0,0,0,0,0
200 FOR J = 1 TO Q
210 FOR I = 1 TO N
220 READ C(I, J)
230 NEXT I
240 NEXT J
250 DATA 2,3,3,5,4,4,5,3
260 FOR H = 1 TO N
270 READ T(H)
280 NEXT H
290 DATA 0,200,60,70,56,120,70,210
300 DATA 220,0,300,100,20,130,250,190
310 DATA 60,300,0,80,600,90,100,300
320 DATA 20,350,80,0,60,10,360,30
330 DATA 70,40,600,46,0,150,80,20
340 DATA 45,85,90,50,60,0,100,40
350 DATA 180,235,100,520,90,250,0,85
360 DATA 335,185,25,64,40,100,70,0
370 FOR I = 1 TO N
380 FOR J = 1 TO N
390 READ P(I, J)
400 NEXT J
```

```
410 NEXT I
420 DATA 24,27,0,0,0
430 DATA 21,24,0,0,0
440 DATA 21,21,15,0,0
450 DATA 21,84,0,0,0
460 DATA 84,12,0,0,0
470 DATA 12,42,0,0,0
480 DATA 27,15,42,15,0
490 DATA 15,0,0,0,0
500 FOR I = 1 TO N
510 J = 1
520 H = C(I, J)
530 FOR J = 2 TO Q
540 K = C(I, J)
550 READ L(H, K)
560 L(K, H) = L(H, K)
570 NEXT J
580 NEXT I
590 FOR H = 1 TO N
600 FOR K = 1 TO N
610 IF K <= H GOTO 1550
630 B = 1
640 R = 1
650 I = H
660 B5 = 1
670 N(B, R) = I
680 L3(B, R) = 0
690 L1(B) = 1
700 L0 = 0
710 R = R + 1
720 K5 = B5
730 FOR B = 1 TO K5
740 X = 0
750 IF K5 = 1 GOTO 770
760 IF L1(B) = 0 GOTO 1240
770 I = N(B, R - 1)
780 Y = 1
790 FOR J = 2 TO Q
800 C1 = C(I, J)
810 IF C1 = 0 GOTO 1220
820 IF C1 = K GOTO 840
830 GOTO 1040
```

```
840 IF Y = 1 THEN B1 = B
850 IF Y = 1 GOTO 910
860 B1 = B5 + 1
870 B5 = B1
880 FOR R2 = 1 TO R - 1
890 N(B1, R2) = N(B, R2)
900 NEXT R2
910 N(B1, R) = C1
920 L3(B1, R) = L3(B, R - 1) + L(I, C1)
930 L1(B1) = 0
940 IF L0 = 0 GOTO 960
950 IF L3(B1, R) > L0 GOTO 1240
960 L0 = L3(B1, R)
970 FOR R2 = 1 TO R
980 N2(R2) = N(B1, R2)
990 NEXT R2
1000 FOR R2 = R + 1 TO N
1010 N2(R2) = 0
1020 NEXT R2
1030 GOTO 1220
1040 IF R = 2 GOTO 1080
1050 FOR R2 = R - 1 TO 1 STEP -1
1060 IF C1 = N(B, R2) GOTO 1210
1070 NEXT R2
1080 IF Y = 1 THEN B1 = B
1090 IF Y = 1 GOTO 1150
1100 B1 = B5 + 1
1110 B5 = B1
1120 FOR R2 = 1 TO R - 1
1130 N(B1, R2) = N(B, R2)
1140 NEXT R2
1150 L1(B1) = 1
1160 N(B1, R) = C1
1170 L3(B1, R) = L3(B, R - 1) + L(I, C1)
1180 IF L0 = 0 GOTO 1200
1190 IF L3(B1, R) > L0 THEN L1(B1) = 0
1200 Y = Y + 1
1210 NEXT J
1220 IF Y = 1 THEN L1(B) = 0
1230 X = X + 1
1240 NEXT B
1250 IF X = 0 GOTO 1270
```

```
1260 GOTO 710
1270 M = M5 + 1
1280 M5 = M
1290 L4(M) = L0
1300 H4(M) = H
1310 K4(M) = K
1320 W(M) = 0
1330 FOR R = 1 TO N
1340 N4(M, R) = N2(R)
1350 IF N2(R) > 0 THEN X2 = R
1360 NEXT R
1370 IF X2 < 3 GOTO 1550
1380 T1 = 0
1390 FOR R2 = 2 TO N - 1
1400 IF N2(R2 + 1) = 0 GOTO 1470
1410 N3 = N2(R2)
1420 IF T1 = 0 THEN T1 = T(N3)
1430 IF T1 = T(N3) GOTO 1460
1440 IF T1 < T(N3) GOTO 1460
1450 T1 = T(N3)
1460 NEXT R2
1470 IF P(H, K) > P(K, H) THEN P8 = P(H, K)
1480 IF P(H, K) > P(K, H) GOTO 1500
1490 P8 = P(K, H)
1500 T0 = C0 * Q0 * T8 / P0 / P8
1510 IF T0 < T1 THEN F = F + 1
1520 IF T0 < T1 THEN M(F) = M
1530 IF T0 < T1 THEN F5 = F
1540 IF T0 < T1 THEN W(M) = 1
1550 NEXT K
1560 NEXT H
1570 FOR I = 1 TO N
1580 FOR J = 1 TO Q
1590 C1(I, J) = C(I, J)
1600 NEXT J
1610 NEXT I
1620 FOR F = 1 TO F5
1630 M = M(F)
1640 I = H4(M)
1650 C1(I, 1) = -1
1660 FOR R = 2 TO N
1670 IF N4(M, R) = 0 GOTO 1760
```

Продолжение приложения 2

```
1680 I1 = N4(M, R)
1690 C1(I1, 1) = -1
1700 FOR J = 2 TO Q
1710 IF C(I, J) = I1 THEN C1(I, J) = -1
1720 IF C(I, J) = I1 THEN I = I1
1730 IF C(I, J) = I1 GOTO 1750
1740 NEXT J
1750 NEXT R
1760 FOR R = N TO 2 STEP -1
1770 IF N4(M, R) = 0 GOTO 1850
1780 I1 = N4(M, R)
1790 C1(I1, 1) = -1
1800 FOR J = 2 TO Q
1810 IF C1(I1, J) <> N4(M, R - 1) GOTO 1840
1820 C1(I1, J) = -1
1830 GOTO 1850
1840 NEXT J
1850 NEXT R
1860 NEXT F
1870 FOR I = 1 TO N
1880 FOR J = 2 TO Q
1890 IF C1(I, J) = 0 GOTO 2110
1900 IF C1(I, J) = -1 GOTO 2100
1910 H = I
1920 K = C(I, J)
1930 IF K < H THEN K = I
1940 IF K = I THEN H = C(I, J)
1950 IF P(H, K) > P(K, H) THEN P8 = P(H, K)
1960 IF P(H, K) > P(K, H) GOTO 1980
1970 P8 = P(K, H)
1980 S1 = Q0 * T8 / P8
1990 IF S1 > S0 GOTO 2100
2000 FOR M = 1 TO M5
2010 IF H4(M) <> H GOTO 2090
2020 IF K4(M) <> K GOTO 2090
2030 IF W(M) = 1 GOTO 2100
2040 F = F5 + 1
2050 F5 = F
2060 M(F) = M
2070 W(M) = 1
2080 GOTO 2100
2090 NEXT M
```

```
2100 NEXT J
2110 NEXT I
2120 FOR I = 1 TO N
2130 J = 1
2140 X = 0
2150 B = 0
2160 IF C1(I, J) = -1 GOTO 2640
2170 FOR J = 2 TO Q
2180 IF C(I, J) = 0 GOTO 2430
2190 H = I
2200 K = C(I, J)
2210 IF K < H THEN K = I
2220 IF K = I THEN H = C(I, J)
2230 IF P(H, K) > P(K, H) THEN P8 = P(H, K)
2240 IF P(H, K) > P(K, H) GOTO 2260
2250 P8 = P(K, H)
2260 S = Q0 * T8 / P8
2270 IF S > S0 GOTO 2390
2280 FOR M = 1 TO M5
2290 IF H4(M) <> H GOTO 2370
2300 IF K4(M) <> K GOTO 2370
2310 IF W(M) = 1 GOTO 2420
2320 F = F5 + 1
2330 F5 = F
2340 X = X + 1
2350 M(F) = M
2360 W(M) = 1
2370 NEXT M
2380 GOTO 2420
2390 B = B + 1
2400 S4(B) = S
2410 N2(B) = I
2420 NEXT J
2430 IF X > 0 GOTO 2640
2440 IF B = 0 GOTO 2640
2450 FOR B = 1 TO B5
2460 IF B = 1 THEN S = S4(B)
2470 IF B = 1 THEN Y = B
2480 IF B = 1 GOTO 2510
2490 IF S > S4(B) GOTO 2510
2500 Y = B
2510 NEXT B
```


Окончание приложения 2

```
2520 F = F5 + 1
2530 F5 = F
2540 J = N2(Y)
2550 H = I
2560 K = C(I, J)
2570 IF K < H THEN K = I
2580 IF K = I THEN H = C(I, J)
2590 FOR M = 1 TO M5
2600 IF H4(M) <> H GOTO 2630
2610 IF K4(M) <> K GOTO 2630
2620 M(F) = M
2630 NEXT M
2640 NEXT I
2650 PRINT
2660 FOR F = 1 TO F5
2665 COLOR 13
2670 PRINT "    МАРШРУТ"; F
2680 M = M(F)
2685 COLOR 12
2690 PRINT "ПУНКТЫ"; H4(M);
2700 FOR R = 2 TO N
2710 IF N4(M, R) = 0 GOTO 2740
2720 PRINT "-"; N4(M, R);
2730 NEXT R
2740 PRINT
2745 COLOR 11
2750 PRINT "ДЛИНА МАРШРУТА"; H4(M); "-"; K4(M); : COLOR 9:
PRINT " L="; L4(M)
2760 PRINT
2770 NEXT F
```

```

uses Crt,Graph;
const
  inter:set of char=['0','1','2','3','4','5','6','7','8','9','-'];
  rea:set of char=['0','1','2','3','4','5','6','7','8','9','-','.'];
var
  a,ch:char;
  N,W,mm,i1,j1:integer;
  S:real;
  grDriver: Integer;
  grMode: Integer;
  ErrCode: Integer;
  P,PP: array[1..8,1..8] of real;
  Optim: array[1..10] of integer;

procedure InInt( var k:integer);
var
  ch:char;
  St:string;
  i,code:word;
begin
  St:="";
  i:=1;
  repeat
    ch:=readkey;
    while (not(ch in inter))and(ch<>#13)and(ch<>#8) do ch:=readkey;
    if (ch<>#13)and(ch<>#8) then
      begin
        write(ch);
        St:=St+ch;
        i:=i+1;
      end;
    if (ch=#8)and(St<>") then
      begin
        Delete(St,length(St),1);
        GotoXY(WhereX-1,WhereY);
        ClrEol;
        i:=i-1;
      end;
  until ((ch=#13)or(i>5))and(St<>");
  val(St,k,code);
  writeln;
end;

```

```

procedure InReal( var k:real);
var
  ch:char;
  St:string;
  i,code:word;
begin
  St:="";
  i:=1;
  repeat
    ch:=readkey;
    while (not(ch in rea))and(ch<>#13)and(ch<>#8) do ch:=readkey;
    if (ch<>#13)and(ch<>#8) then
      begin
        write(ch);
        St:=St+ch;
        i:=i+1;
      end;
    if (ch=#8)and(St<>") then
      begin
        Delete(St,length(St),1);
        GotoXY(WhereX-1,WhereY);
        ClrEol;
        i:=i-1;
      end;
  until ((ch=#13)or(i>7))and(St<>");
  val(St,k,code);
  writeln;
end;

```

```

procedure NLK;
begin
  TextColor(2);
  writeln;
  write('НАЖМИТЕ ЛЮБУЮ КЛАВИШУ');
  ch:=readkey;
  TextColor(15);
end;

```

```

procedure Color(X,Y:integer);
begin
  TextColor(X);

```

```
TextBackGround(Y);
end;
```

```
procedure BarR(Color,X,Y,a,b:integer);
var i,j:word;
begin
  TextColor(Color);
  GoToXY(X,Y);
  write('┌');
  for i:=0 to a-2 do write('=');
  writeln('┐');
  for j:=0 to b-2 do
    begin
      GoToXY(X,WhereY);
      write('│');
      GoToXY(X+a,WhereY);
      writeln('│');
    end;
  GoToXY(X,WhereY);
  write('└');
  for i:=0 to a-2 do write('=');
  writeln('┘');
  TextColor(15);
end;
```

```
procedure RisGraf;
const outm:array[1..8,1..2] of integer=((130,200),(330,90),(450,160),
(370,390),(250,390),(190,110),
(150,320),(450,280));

var
  i,j,l:integer;
  x,y,x1,y1,dx,dy:integer;
  St:string;
begin
  for i:=1 to N do
    begin
      x:=outm[i,1];
      y:=outm[i,2];
      for j:=i+1 to N do
        begin
          if PP[i,j]<>0 then
```

```

begin
  x1:=outm[j,1];
  y1:=outm[j,2];
  SetColor(10);
  Line(x,y,x1,y1);
  dx:=trunc((x1+x)/2)-4;
  dy:=trunc((y1+y)/2)-4;
  SetColor(15);
  Str(PP[i,j]:3:1,St);
  OutTextXY(dx,dy,St);
end;
end;
SetColor(10);
SetFillStyle(1,2);
SetLineStyle(0,0,3);
FillEllipse(x,y,15,15);
Circle(x,y,15);
Str(i,St);
SetColor(15);
SetTextStyle(0,0,2);
OutTextXY(x-6,y-6,St);
SetTextStyle(0,0,1);
SetLineStyle(0,0,1);
end;
if mm=2 then
  begin
    SetLineStyle(0,0,3);
    i:=1;
    l:=1;
    repeat
      j:=Optim[l];
      SetColor(9);
      x1:=outm[j,1];
      y1:=outm[j,2];
      x:=outm[i,1];
      y:=outm[i,2];
      Line(x,y,x1,y1);
      SetColor(10);
      dx:=trunc((x1+x)/2)-4;
      dy:=trunc((y1+y)/2)-4;
      SetColor(15);
      Str(PP[i,j]:3:1,St);

```

```

    OutTextXY(dx,dy,St);
    SetColor(10);
    SetFillStyle(1,2);
    FillEllipse(outm[i,1],outm[i,2],15,15);
    Str(i,St);
    SetColor(15);
    SetTextStyle(0,0,2);
    OutTextXY(outm[i,1]-6,outm[i,2]-6,St);
    SetTextStyle(0,0,1);
    SetColor(10);
    i:=j;
    l:=l+1;
until j=N;
    SetColor(10);
    SetFillStyle(1,2);
    FillEllipse(outm[N,1],outm[N,2],15,15);
    Str(i,St);
    SetColor(15);
    SetTextStyle(0,0,2);
    OutTextXY(outm[N,1]-6,outm[N,2]-6,St);
    SetTextStyle(0,0,1);
    SetColor(10);
    SetLineStyle(0,0,1);

end; { if }
end;

procedure OutGraf;
begin
    grDriver:=Detect;
    InitGraph(grDriver, grMode,"");
    ErrCode := GraphResult;
    if ErrCode = grOk then
        begin { Do graphics }
            RisGraf;
            Delay(3000);
            SetTextStyle(0,0,2);
            SetColor(10);
            if mm<>2 then
                OutTextXY(130,30,'ГРАФ ТРАНСПОРТНОЙ СЕТИ ')
            else

```

Продолжение приложения 3

```

    OutTextXY(130,30,'КРАТЧАЙШИЙ ПУТЬ В ГРАФЕ');
    SetTextStyle(3,0,1);
    OutTextXY(50,50,'=====');
    SetColor(15);
    OutTextXY(100,450,'ДЛЯ ПРОДОЛЖЕНИЯ РАБОТЫ
                      НАЖМИТЕ ЛЮБУЮ КЛАВИШУ');

    ch:=readkey;
    CloseGraph;
    ClrScr;
end
else
    Writeln('Graphics error:',
            GraphErrorMsg(ErrCode));
end;

procedure Matriz;
var
    i,j:integer;
    xx,yy:integer;
begin
    GoToXY(14,4);
    TextColor(10);
    Write('МАТРИЦА, ОПРЕДЕЛЯЮЩАЯ СВЯЗИ МЕЖДУ ВЕРШИНАМИ В
          ГРАФЕ');
    GoToXY(14,5);
    Write('=====');
    xx:=40-(trunc(N/2))*7;
    GoToXY(xx,7);
    for j:=1 to N do
        begin
            for i:=1 to N do
                begin
                    if P[i,j]=0 then TextColor(15) else TextColor(10);
                    GoToXY(xx+(i-1)*7,WhereY);
                    write(P[i,j]:4:1);
                end;
            writeln;
            GoToXY(xx,WhereY);
        end;
    end;
end;

```

```

procedure Connect;
var
  i,j,l,xx,yy:integer;
  dl:real;
  St:string;
begin
  Window(1,1,80,25);
  i:=1;
  repeat { по i }
    j:=1;
    repeat { по j }
      ClrScr;
      Matriz;
      BarR(2,5,19,40,4);
      GoToXY(5,25);
      TextColor(15);
      Write(' Конец ввода для данной вершины – клавиша "0"');
      GoToXY(10,20);
      TextColor(10);
      Write(' ВЕРШИНА '); TextColor(15); Write(i);
      TextColor(10); Write(' СВЯЗАНА С ВЕРШИНОЙ ');
      xx:=WhereX;
      yy:=WhereY;
      TextColor(15);
      repeat
        GotoXy(xx,yy);
        Write(' ');
        GoToXY(xx,yy);
        InInt(1);
      until (l>=0)and(l<=N)and(l<>i) ;
      if l<>0 then
        begin
          GoToXY(10,22); TextColor(10);
          write('ДЛИНА ДУГИ '); TextColor(15); Write(i,' <--> ',l);
          TextColor(10); write(' ГРАФА ');
          xx:=WhereX;
          yy:=WhereY;
          TextColor(15);
          repeat
            GotoXy(xx,yy);
            Write(' ');

```



```

    GoToXY(xx,yy);
    InReal(dl);
until (dl>0);
P[i,l]:=dl; P[l,i]:=dl;
Delay(700);
GoToXY(10,22);
Write('          ');
j:=j+1;
end; {if};
until (j=N) or (l=0); {j}
i:=i+1;
until (i=N); {i}
PP:=P;
end;

```

```

procedure Rasz;
var
  I,K,xx,j,l:integer;
  S1:real;
procedure RR;
begin
  while(P[I,W]=0)and(I<N) do I:=I+1; {*****}
if I<>N then
begin
  S:=P[I,W]+P[I,N];
  K:=I;
  I:=I+1;
repeat
  if P[I,W]<>0 then begin
    S1:=P[I,W]+P[I,N];
    if S1<S then begin S:=S1;
      K:=I;
    end;
  end;

  I:=I+1;
until I>N;
end {if}
else if P[I,W]<>0 then S:=P[I,W] else S:=10000; {*****}
end;
begin

```

```

for W:=N-1 downto 1 do
begin
  I:=W+1;
  RR;
  if K=0 then K:=N;
  for I:=W to N do if I<>K then
    P[I,W]:=0; { ***** }
  P[W,N]:=S;
  ClrScr;
  TextColor(10);
  GoToXY(14,2);
  Write('ШАГ ПРОЦЕССА ОПТИМИЗАЦИИ ПУТИ В ГРАФЕ : ');
  TextColor(15); write('W=',W);
  TextColor(10);
  GoToXY(14,3);
  if S<10000 then
  begin
    Write('ДЛИНА КРАТЧАЙШЕГО ПУТИ НА ДАННОМ ШАГЕ : ');
    TextColor(15); write('S=',S:3:1);
    TextColor(10)
  end
  else
    Writeln("ДЛИНА КРАТЧАЙШЕГО ПУТИ НА ДАННОМ ШАГЕ : ---");
    GoToXY(1,5);
    GoToXY(14,6);
    TextColor(10);
    Write(' МАТРИЦА ОПТИМИЗАЦИИ ПУТИ В ГРАФЕ НА ДАННОМ
ШАГЕ');
    GoToXY(14,7);
    Write('=====Í');
    xx:=40-(trunc(N/2))*7;
    GoToXY(xx,8);
    for j:=1 to N do
    begin
      for i:=1 to N do
      begin
        if P[i,j]=0 then TextColor(15) else TextColor(10);
        GoToXY(xx+(i-1)*7,WhereY);
        { ***** } if P[i,j]<10000 then write(P[i,j]:4:1) else write(' ---');
      end;
    end;
    writeln;

```

```

    GoToXY(xx,WhereY);
end;
GoToXY(14,24);
NLK;
GoToXY(1,WhereY);
ClrEol;
end; {for W}
j:=1; l:=1;
TextColor(10);
GoToXY(10,20);
if P[1,N]<10000 then
begin
write('ОПТИМАЛЬНЫЙ ВАРИАНТ ПУТИ В ГРАФЕ : ');
TextColor(15); write('1');
i:=j+1;
repeat
while (P[i,j]=0)and(i<>N) do i:=i+1;
j:=i;
Optim[l]:=j;
l:=l+1;
TextColor(10); write('-->');
TextColor(15); write(j);
until j=N;
xx:=WhereX;
TextColor(10);
GoToXY(10,22);
write('ДЛИНА ОПТИМАЛЬНОГО ПУТИ ');
TextColor(15); writeln('T=',P[1,N]:3:1);
TextColor(10);
BarR(10,5,19,xx,4);
mm:=2;
end else
begin
writeln('ГРАФ НЕ ЯВЛЯЕТСЯ СВЯЗНЫМ');
GoToXY(10,WhereY);
write('В ДАННОМ ГРАФЕ НЕ СУЩЕСТВУЕТ СВЯЗИ ');
TextColor(15); write('1 ');
TextColor(10); write('<---> ');
TextColor(15); writeln(N);
TextColor(10);
BarR(10,5,19,55,3);

```

```

        mm:=3;
    end; {else}
NLK;
end;

procedure start;
var
    code:word;
begin
    a:='0';
    repeat
        for i1:=1 to 8 do
            for j1:=0 to 8 do P[i1,j1]:=0;
        ClrScr;
        window(12,7,75,25);
        TextColor(10);
        writeln;
        writeln('    ПРОГРАММА ОПРЕДЕЛЕНИЯ КРАТЧАЙШЕГО ПУТИ ');
        writeln('        В ОРИЕНТИРОВАННОМ ГРАФЕ МЕТОДОМ ');
        writeln('        ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ');
        BarR(10,5,1,45,4);
        TextColor(15);
        Delay(2000);
        NLK;
        GoToXY(1,WhereY);
        ClrEol;
        writeln;
        TextColor(10);
        GotoXY(3,10);
        write(' ВВЕДИТЕ КОЛИЧЕСТВО ВЕРШИН ГРАФА (4...8) : ');
        repeat
            GotoXy(48,10);
            ClrEol;
            InInt(N);
        until (N>3)and(N<=8);
        ClrScr;
        Connect;
        mm:=1;
        OutGraf;
        Rasz;
        OutGraf;
    end;
end;

```

```
TextColor(15);
BarR(10,20,8,30,4);
GoToXY(27,10);
TextColor(15);
Write('ПОВТОРИТЬ? Д/Н');
ch:=readkey;
until (ch<>'Д')and(ch<>'д')and(ch<>'I')and(ch<>'L')
end;

BEGIN
  Color(15,0);
  start;
END.
```

```

uses Crt,Graph;
const
  inter:set of char=['0','1','2','3','4','5','6','7','8','9'];
  C:array[1..8,1..6] of integer=
    ((1,2,7,0,0,0),(2,3,1,0,0,0),(3,2,4,7,0,0),(4,3,5,0,0,0),
     (5,4,6,0,0,0),(6,5,7,0,0,0),(7,1,3,6,8,0),(8,7,0,0,0,0));
  Tconst:array[1..8] of integer=(2,3,3,5,4,4,5,3);
  Pconst:array[1..8,1..8] of integer=
    ((0,220,60,20,70,45,180,335),(200,0,56,350,40,85,235,185),
     (60,300,0,80,600,90,100,25),(70,350,240,0,46,50,520,64),
     (56,20,540,60,0,60,90,40),(120,130,105,10,150,0,250,100),
     (70,250,30,360,80,100,0,70),(210,190,110,30,20,40,85,0));
  Lconst:array [1..40] of integer=
    (24,27,0,0,0,21,24,0,0,0,21,21,15,0,0,21,84,0,0,0,
     84,12,0,0,0,12,42,0,0,0,27,15,42,15,0,15,0,0,0,0);
var
  ch:char;
  grDriver,grMode,ErrCode: Integer;
  Q,N,Q0,S0,T8:integer;
  C0,P0:real;
  P:array[1..8,1..8] of integer;
  T:array[1..8] of integer;
  L:array[1..8,1..8] of integer;
  Nm,N4,L3:array[1..30,1..8] of integer;
  H4,K4,L4,W,Mm:array[1..50] of integer;
  S4:array[1..10] of real;
  N2:array[1..10] of integer;
  L1:array[1..20] of integer;
  C1m:array[1..8,1..6] of integer;
  h,k,b,r,i,b5,L0,k5,y,j,x,p8,x2,C1,b1,r2,m,m5,t1,n3,f,f5:integer;
  t0,S1,S:real;
  xx,yy:integer;

procedure InInteger( var k:integer);
var
  ch:char;
  St:string;
  i,code:word;
begin
  St:="";
  i:=1;

```

```

repeat
ch:=readkey;
while (not(ch in inter))and(ch<>#13)and(ch<>#8) do ch:=readkey;
if (ch<>#13)and(ch<>#8)and(i<=3) then
begin
write(ch);
St:=St+ch;
i:=i+1;
end;
if (ch=#8)and(St<>") then
begin
Delete(St,length(St),1);
GotoXY(WhereX-1,WhereY);
write(' ');
GotoXY(WhereX-1,WhereY);
i:=i-1;
end;
val(St,k,code);
until (ch=#13)and(St<>")and(k<>0);
val(St,k,code);
end;

```

```

procedure InIntegerT( var k:integer);
var
ch:char;
St:string;
i,code:word;
begin
St:="";
i:=1;
repeat
ch:=readkey;
while (not(ch in inter))and(ch<>#13)and(ch<>#8) do ch:=readkey;
if (ch<>#13)and(ch<>#8)and(i<=2) then
begin
write(ch);
St:=St+ch;
i:=i+1;
end;
if (ch=#8)and(St<>") then
begin

```

```

Delete(St,length(St),1);
GotoXY(WhereX-1,WhereY);
write(' ');
GotoXY(WhereX-1,WhereY);
i:=i-1;
end;
until (ch=#13)and(St<>"");
val(St,k,code);
end;

```

```

procedure NLK;
begin
  TextColor(2);
  writeln;
  write('НАЖМИТЕ ЛЮБУЮ КЛАВИШУ');
  ch:=readkey;
  TextColor(15);
end;

```

```

procedure Color(X,Y:integer);
begin
  TextColor(X);
  TextBackGround(Y);
end;

```

```

procedure BarR(Color,X,Y,a,b:integer);
var i,j:word;
begin
  TextColor(Color);
  GoToXY(X,Y);
  write('┌');
  for i:=0 to a-2 do write('=');
  writeln('┐');
  for j:=0 to b-2 do
    begin
      GoToXY(X,WhereY);
      write('│');
      GoToXY(X+a,WhereY);
      writeln('│');
    end;
  GoToXY(X,WhereY);

```



```

write('L');
for i:=0 to a-2 do write('=');
writeln('L');
TextColor(15);
end;

```

```

procedure Setzn;
var i,j,k,ll,h:integer;
begin
  Q:=6; N:=8; C0:=0.5;
  Q0:=40; S0:=12; T8:=60;
  P0:=1.1;
  m5:=0;
  m:=0;
  ll:=1;
  f:=0;
  for i:=1 to 8 do
    for j:=1 to 8 do
      P[i,j]:=Pconst[i,j];
    for i:=1 to 8 do
      T[i]:=Tconst[i];
    for i:=1 to N do
      begin
        j:=1;
        h:=C[i,j];
        for j:=2 to Q do
          begin
            k:=C[i,j];
            if (h<>0)and(k<>0) then L[h,k]:=Lconst[ll];
            ll:=ll+1;
            if (h<>0)and(k<>0) then L[k,h]:=L[h,k]
          end;
        end;
      end;
    end;
  end;
end;

```

```

procedure Pr11;
var mmm:string;
begin
  i:=Nm[b,r-1];
  y:=1;
  for j:=2 to Q do

```

```

begin
  C1:=C[i,j];
  if C1=0 then begin
    if y=1 then L1[b]:=0;
    x:=x+1;
    exit;
  end;
  if C1=k then
  begin {840}
    if y=1 then b1:=b else
      begin
        b1:=b5+1;
        b5:=b1;
        for r2:=1 to r-1 do
          Nm[b1,r2]:=Nm[b,r2];
        end;
        Nm[b1,r]:=C1;
        L3[b1,r]:=L3[b,r-1]+L[i,C1];
        L1[B1]:=0;
        if (L0<>0)and(L3[B1,R]>L0) then exit;
        L0:=L3[B1,R];
        for r2:=1 to r do
          N2[r2]:=Nm[b1,r2];
        for r2:=r+1 to N do
          N2[r2]:=0;
        end {840}
      else
        begin
          {c 1040}
          mmm:="";
          if R<>2 then
            begin
              r2:=r-1;
              while (r2>=1)and(mmm="") do {!!!!!!!!!!!!!!!!!!!!}
                begin if C1=Nm[b,r2] then mmm:='quit';
                  r2:=r2-1; {!!!!!!!!!!!!!!!!!!!!!!!!!!!!}
                end;
              end;
            if mmm<>'quit' then
              begin
                if y=1 then b1:=b else

```

```

begin
  b1:=b5+1;
  b5:=b1;
  for r2:=1 to r-1 do
    Nm[b1,r2]:=Nm[b,r2];
  end;
  L1[b1]:=1;
  Nm[b1,r]:=C1;
  L3[b1,r]:=L3[b,r-1]+L[i,C1];
  if L0<>0 then
    if L3[b1,r]>L0 then L1[b1]:=0;
  y:=y+1;
end; {quit}
end;
end; {j}
if y=1 then L1[b]:=0;
x:=x+1;
end;

```

```

procedure P710;
begin
  r:=r+1;
  k5:=b5;
  for b:=1 to k5 do
    begin
      x:=0;
      if k5=1 then Pr11
        else if L1[b]<>0 then Pr11;
    end;
    if x<>0 then P710;
  end;
end;

```

```

procedure Pr1;
begin
  b:=1; r:=1; i:=h; b5:=1;
  Nm[b,r]:=i;
  L3[b,r]:=0;
  L1[b]:=1;
  L0:=0;
  P710;
  m:=m5+1;

```

```

m5:=m;
L4[m]:=L0;
H4[m]:=H;
K4[m]:=K;
W[m]:=0;
for r:=1 to N do
begin
  N4[m,r]:=N2[r];
  if N2[r]>0 then x2:=r
end;
if x2<3 then exit;
t1:=0;
r2:=2;
while (r2<=N-1)and(N2[r2+1]<>0) do {!!!!!!!!!!!!or!!!!!!!!!!!!!!!!!!!!!!}
begin
  N3:=N2[r2];
  if t1=0 then t1:=T[N3];
  if t1>T[N3] then t1:=t[N3];
  r2:=r2+1;
end;
if P[h,k]>P[k,h] then P8:=P[h,k]
  else P8:=P[k,h];
t0:=c0*Q0*t8/p0/p8;
if t0<t1 then begin
  f:=f+1;
  Mm[f]:=M;
  f5:=f;
  W[m]:=1;
end;
end;

procedure P1570;
var i1,ind:integer;
    mmm:string;
begin
  for i:=1 to N do
    for j:=1 to Q do
      C1m[i,j]:=C[i,j];
  for f:=1 to f5 do
    begin
      M:=Mm[f];

```

```

i:=H4[m];
C1m[i,1]:=-1;
r:=2;
while (r<=N)and(N4[m,r]<>0) do  {!!!!!!!!!!!!!!!!!!!!!!or}
begin
  i1:=N4[m,r];
  C1m[i1,1]:=-1;
  j:=2;
  while (j<=Q)and(C[i,j]<>i1) do j:=j+1;  {!!!!!!!!!!or}
  if (j<=Q)and(C[i,j]=i1) then
    begin
      C1m[i,j]:=-1;
      i:=i1;
    end;
  r:=r+1;
end; {r}
for r:=N downto 2 do
  if N4[m,r]<>0 then
    begin
      i1:=N4[m,r];
      C1m[i1,1]:=-1;
      j:=2;
      while (j<=Q)and(C1m[i1,j]<>N4[m,r-1]) do j:=j+1;  {!!!!!!or}
      if (j<=Q)and(C1m[i1,j]=N4[m,r-1]) then C1m[i1,j]:=-1;
    end; {r}
end; {f}
for i:=1 to N do
begin
  for j:=2 to Q do
  begin
    mmm:="";
    if (C1m[i,j]=0)or(C1m[i,j]=-1) then mmm:='nextj'; {1!!!!!!!!!!}
    if mmm="" then
      begin
        h:=i;
        k:=C[i,j];
        if k<h then k:=i;
        if k=i then h:=C[i,j];
        if P[h,k]>P[k,h] then P8:=P[h,k]
          else P8:=P[k,h];
        if p8<>0 then  S1:=Q0*t8/P8;

```

```

if S1>S0 then mmm:='nextj';
if mmm="" then
begin
m:=1;
ind:=0;
while (m<=m5)and(ind=0) do
begin
if (H4[m]=h)and(K4[m]=k) then
if W[m]=1 then ind:=1
else begin
f:=f5+1;
f5:=f;
Mm[f]:=m;
W[m]:=1;
ind:=1;
end;
m:=m+1;
end; {while}
end;
end;{mmm}
end;{j}
end;{i}
for i:=1 to N do
begin
j:=1;
x:=0;
b:=0;
if C1m[i,j]<>-1 then
begin
j:=2;
while (j<=Q)and(C[i,j]<>0) do
begin
h:=i;
k:=C[i,j];
if k<h then k:=i;
if k=i then h:=C[i,j];
if P[h,k]>P[k,h] then P8:=P[h,k]
else P8:=P[k,h];
S:=Q0*t8/P8;
if S>S0 then
begin

```

```

b:=b+1;
S4[b]:=S;
N2[b]:=i;
end else
begin
  m:=1;
  ind:=0;
  while (m<=m5)and(ind=0) do
  begin
    if (H4[m]=h)and(K4[m]=k) then
      if W[m]=1 then ind:=1
      else begin
        f:=f5+1;
        f5:=f;
        x:=x+1;
        Mm[f]:=m;
        W[m]:=1;
      end;
      m:=m+1;
    end; { while }
  end; { else }
j:=j+1;
end; { while j }
if (x<=0)and(b<>0) then
begin
  for b:=1 to b5 do
    if b=1 then begin
      S:=S4[b];
      y:=b
    end else if S<=S4[b] then y:=b;
  f:=f5+1;
  f5:=f;
  j:=N2[y];
  h:=i;
  k:=C[i,j];
  if k<h then k:=i;
  if k=i then h:=C[i,j];
  for m:=1 to m5 do
    if (H4[m]=h)and(K4[m]=k) then Mm[f]:=m;
  end; { if }
end; { if }

```

```

    end; {i}
end;

```

```

procedure Tabl;
begin
    TextColor(10);
    writeln('          СХЕМА АВТОБУСНЫХ МАРШРУТОВ');
    writeln('===== ');
    TextColor(2);
    writeln(' ');
    writeln('      НОМЕР      ПУНКТЫ      ВРЕМЯ ');
    writeln('      МАРШРУТА      ДВИЖЕНИЯ ');
    write('


|                   |        |                   |
|-------------------|--------|-------------------|
| НОМЕР<br>МАРШРУТА | ПУНКТЫ | ВРЕМЯ<br>ДВИЖЕНИЯ |
|-------------------|--------|-------------------|


    ');
end;

```

```

procedure Stroka;
begin
    writeln('


|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|


    ');
    writeln(' ');
    write(' ');
end;

```

```

procedure PrintRes;
var ff:integer;
procedure Wiw(Col:integer);
begin
    TextColor(2);
    if WhereY>20 then begin ClrScr; ff:=1;end;
    GoToXY(1,6+2*(ff-1));
    Stroka;
    GoToXY(4,7+2*(ff-1));
    TextColor(Col);
    write(f);
    TextColor(2);
    m:=Mm[f];
    write(' (',H4[m], ' ---',K4[m],')');
    TextColor(Col);
    GoToXY(20,7+2*(ff-1));
    write(H4[m]);
    r:=2;
    while (r<=N)and(N4[m,r]<>0) do

```



```

    begin
      write(' — ',N4[m,r]);
      r:=r+1;
    end;
    GoToXY(49,7+2*(ff-1));
    write(L4[m],' мин. ');
  end; {Wiw}
begin
  ClrScr;
  window(10,3,75,25);
  TextColor(15);
  GoToXY(1,23);
  Write('НАЖМИТЕ ЛЮБУЮ КЛАВИШУ');
  GoToXY(1,1);
  Tabl;
  ff:=1;
  for f:=1 to f5 do
    begin
      if Mm[f]<>0 then
        begin
          Wiw(15);
          ch:=readkey;
          Wiw(10);
          ff:=ff+1;
        end;
      end;
    end;

  procedure Rasch;
  begin
    for h:=1 to N do
      for k:=1 to N do
        if k>h then Pr1;
      P1570;
      Printres;
    end;

  procedure OutL;
  procedure Wiw(Col:integer);
  begin
    TextColor(Col);

```

```

GoToXY(11+(xx-1)*7,12+(yy-1));
if L[xx,yy]<>0 then write(L[xx,yy])else write('——');
end;
begin
ClrScr;
GoToXY(1,3);
TextColor(10);
writeln(' МАССИВ, ХАРАКТЕРИЗУЮЩИЙ ТРАНСПОРТНУЮ СЕТЬ');
writeln(' ===== ');
TextColor(2);
writeln(' время движения автобусов по участкам транспортной сети в мин. ');
GoToXY(1,7);
writeln('


| Пункты<br>отправ-<br>ления | Пункты прибытия |   |   |   |   |   |   |   |
|----------------------------|-----------------|---|---|---|---|---|---|---|
|                            | 1               | 2 | 3 | 4 | 5 | 6 | 7 | 8 |


');
writeln('


|      |  |  |  |  |  |  |  |  |
|------|--|--|--|--|--|--|--|--|
| 'i,' |  |  |  |  |  |  |  |  |
|------|--|--|--|--|--|--|--|--|


');
Color(10,0);
for i:=1 to 8 do
begin
for j:=1 to 8 do
begin
GoToXY(11+(i-1)*7,12+(j-1));
if L[i,j]<>0 then write(L[i,j])
else write('——');
end;
end;
end;
TextColor(15);
GoToXY(1,25);
write('СТРЕЛКИ');
TextColor(10);write('-переход ');
TextColor(15);write('ЗАБОЙ,ENTER');
TextColor(10);write('-ввод числа ');
TextColor(15);write('ПРОБЕЛ');
TextColor(10);write('-продолжение работы');
{возможность изменения данных}
GoToXY(11,12);
xx:=1;yy:=1;

```

```

Wiw(15);
repeat
ch:=readkey;
case ch of
  #77{вправо}: if xx<8 then
    begin
      Wiw(10);
      xx:=xx+1;
      Wiw(15);
    end;
  #75{влево} : if xx>1 then
    begin
      Wiw(10);
      xx:=xx-1;
      Wiw(15);
    end;
  #80{вниз} : if yy<8 then
    begin
      Wiw(10);
      yy:=yy+1;
      Wiw(15);
    end;
  #72{вверх} : if yy>1 then
    begin
      Wiw(10);
      yy:=yy-1;
      Wiw(15);
    end;
  #8{ввод} : if L[xx,yy]<>0 then
    begin
      TextColor(15);
      GoToXY(11+(xx-1)*7,12+(yy-1));
      write(' ');
      GoToXY(11+(xx-1)*7,12+(yy-1));
      InIntegerT(L[xx,yy]);
      L[yy,xx]:=L[xx,yy];
      GoToXY(11+(xx-1)*7,12+(yy-1));
      write(' ');
      Wiw(10);
      {изменение L[yy,xx]}
      GoToXY(11+(yy-1)*7,12+(xx-1));

```

```

        write(' ');
        TextColor(10);
        GoToXY(11+(yy-1)*7,12+(xx-1));
        if L[xx,yy]<>0 then write(L[xx,yy])else write('——');
        if xx<>8 then xx:=xx+1 else xx:=xx-1;
        Wiw(15);
    end;
end; {case}
until ch=#32; {пробел}
Wiw(10);
GoToXY(1,25);
ClrEol;
TextColor(15);
GoToXY(1,25);
Write('ДЛЯ ПРОДОЛЖЕНИЯ НАЖМИТЕ ЛЮБУЮ КЛАВИШУ');
ch:=readkey;
end;

```

```

procedure RisGraf;
const outm:array[1..8,1..2] of integer=((100,240),(180,90),(300,160),
(430,90),(500,330),(350,390),
(250,300),(150,380));

var
    i,j:integer;
    x,y,x1,y1,dx,dy:integer;
    St:string;
begin
    for i:=1 to N do
        begin
            x:=outm[i,1];
            y:=outm[i,2];
            for j:=i+1 to 8 do
                begin
                    if L[i,j]<>0 then
                        begin
                            x1:=outm[j,1];
                            y1:=outm[j,2];
                            SetColor(9);
                            SetLineStyle(0,0,3);
                            Line(x,y,x1,y1);
                            dx:=trunc((x1+x)/2)-4;

```

```

dy:=trunc((y1+y)/2)-4;
SetColor(15);
Str(L[i,j],St);
OutTextXY(dx,dy,St);
end;
end;
SetColor(10);
SetFillStyle(1,2);
SetLineStyle(0,0,3);
FillEllipse(x,y,15,15);
Circle(x,y,15);
Str(i,St);
SetColor(15);
SetTextStyle(0,0,2);
OutTextXY(x-6,y-6,St);
SetTextStyle(0,0,1);
SetLineStyle(0,0,1);
end;
SetTextStyle(0,0,2);
SetColor(10);
OutTextXY(130,30,'ГРАФ ТРАНСПОРТНОЙ СЕТИ');
SetTextStyle(0,0,1);
OutTextXY(50,50,'=====');
SetColor(15);
OutTextXY(100,450,'ДЛЯ ПРОДОЛЖЕНИЯ РАБОТЫ НАЖМИТЕ
ЛЮБУЮ КЛАВИШУ');
end;

procedure OutTS;
begin
  grDriver:=Detect;
  InitGraph(grDriver, grMode,"");
  ErrCode := GraphResult;
  if ErrCode = grOk then
    begin { Do graphics }
      RisGraf;
      Delay(3000);
      ch:=readkey;
      CloseGraph;
      ClrScr;
    end
end

```

```

else
    Writeln('Graphics error:',
            GraphErrorMsg(ErrCode));
end;

Procedure OutP;
procedure Wiw(Col:integer);
begin
    TextColor(Col);
    GoToXY(11+(xx-1)*7,9+(yy-1));
    if P[xx,yy]<>0 then write(P[xx,yy])else write('——');
end;
begin
ClrScr;
Color(10,0);
writeln('КОРРЕСПОНДЕНЦИЯ ПАССАЖИРОПОТОКОВ В ЧАС ПИК');
Color(2,0);
writeln('
');


|                   |                     |   |   |   |   |   |   |   |
|-------------------|---------------------|---|---|---|---|---|---|---|
| Пункты<br>отправ. | Пассажиропоток, чел |   |   |   |   |   |   |   |
|                   | Пункты прибытия     |   |   |   |   |   |   |   |
|                   | 1                   | 2 | 3 | 4 | 5 | 6 | 7 | 8 |


        writeln('
');
for i:=1 to 8 do
    writeln('
');
    writeln('
');
Color(10,0);
for i:=1 to 8 do
begin
    for j:=1 to 8 do
begin
GoToXY(11+(i-1)*7,9+(j-1));
if P[i,j]<>0 then write(P[i,j])
else write('——');
end;
end;
TextColor(15);
GoToXY(1,25);
write('СТРЕЛКИ');
TextColor(10);write('-переход ');

```

```

TextColor(15);write('ЗАБОЙ,ENTER');
TextColor(10);write('-ввод числа');
TextColor(15);write('ПРОБЕЛ');
TextColor(10);write('-продолжение работы');
{возможность изменения данных}
GoToXY(11,9);
xx:=1;yy:=1;
Wiw(15);
repeat
ch:=readkey;
case ch of
  #77{вправо}: if xx<8 then
    begin
      Wiw(10);
      xx:=xx+1;
      Wiw(15);
    end;
  #75{влево} : if xx>1 then
    begin
      Wiw(10);
      xx:=xx-1;
      Wiw(15);
    end;
  #80{вниз} : if yy<8 then
    begin
      Wiw(10);
      yy:=yy+1;
      Wiw(15);
    end;
  #72{вверх} : if yy>1 then
    begin
      Wiw(10);
      yy:=yy-1;
      Wiw(15);
    end;
  #8{ввод} : if P[xx,yy]<>0 then
    begin
      TextColor(15);
      GoToXY(11+(xx-1)*7,9+(yy-1));
      write(' ');
      GoToXY(11+(xx-1)*7,9+(yy-1));
    end;
end;

```

```

        InInteger(P[xx,yy]);
        GoToXY(11+(xx-1)*7,9+(yy-1));
        write(' ');
        Wiw(10);
        if xx<>8 then xx:=xx+1 else xx:=xx-1;
        Wiw(15);
    end;
end; {case}
until ch=#32; {пробел}
Wiw(10);
end;

procedure OutT;
    procedure Wiw(Col:integer);
    begin
        TextColor(Col);
        GoToXY(12+(xx-1)*7,22);
        write(T[xx]);
    end;
begin
    GoToXY(1,18);
    Color(10,0);
    writeln(' ВРЕМЯ, ЗАТРАЧИВАЕМОЕ ПАССАЖИРАМИ НА
ПЕРЕСАДКИ');
    Color(2,0);
    writeln('


|         |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|
| Пункт   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| t, мин. |   |   |   |   |   |   |   |   |


');
    writeln('


|         |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|
| Пункт   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| t, мин. |   |   |   |   |   |   |   |   |


');
    writeln('


|         |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|
| Пункт   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| t, мин. |   |   |   |   |   |   |   |   |


');
    writeln('


|         |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|
| Пункт   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| t, мин. |   |   |   |   |   |   |   |   |


');
    Color(10,0);
    for i:=1 to 8 do
    begin
        GoToXY(12+(i-1)*7,22);
        write(T[i]);
    end;
    {возможность изменения данных}
    GoToXY(12,22);
    xx:=1;
    Wiw(15);
    repeat

```



```

ch:=readkey;
case ch of
  #77{ВПРАВО}: if xx<8 then
    begin
      Wiw(10);
      xx:=xx+1;
      Wiw(15);
    end;
  #75{ВЛЕВО} : if xx>1 then
    begin
      Wiw(10);
      xx:=xx-1;
      Wiw(15);
    end;
  #8{ВВОД} : begin
    TextColor(15);
    GoToXY(12+(xx-1)*7,22);
    write(' ');
    GoToXY(12+(xx-1)*7,22);
    InIntegerT(T[xx]);
    GoToXY(12+(xx-1)*7,22);
    write(' ');
    Wiw(10);
    if xx<>8 then xx:=xx+1 else xx:=xx-1;
    Wiw(15);
  end;
end;
until ch=#32; {пробел}
Wiw(10);
GoToXY(1,25);
ClrEol;
TextColor(15);
GoToXY(1,25);
Write('ДЛЯ ПРОДОЛЖЕНИЯ НАЖМИТЕ ЛЮБУЮ КЛАВИШУ');
ch:=readkey;
end;

procedure OutData;
begin
  ClrScr;
  window(10,1,75,25);

```

```

TextColor(10);
GoToXY(1,3);
writeln('');
writeln(' ИСХОДНЫЕ ДАННЫЕ ДЛЯ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ ');
writeln('');
writeln;
writeln('Коэффициент неравномерности подхода пассажиров к остановке');
TextColor(15);
writeln('      C0=0.5');
writeln;
TextColor(10);
writeln('Коэффициент внутричасовой неравномерности пассажирского потока');
TextColor(15);
writeln('      P0=1.1');
writeln;
TextColor(10);
writeln('Продолжительность расчетного периода суток');
TextColor(15);
writeln('      T8=60 мин. ');
writeln;
TextColor(10);
writeln('Используемая вместимость автобуса');
TextColor(15);
writeln('      Q0=40 чел. ');
writeln;
TextColor(10);
writeln('Максимальный интервал между отправлениями автобуса');
TextColor(15);
writeln('      Imax=12 мин. ');
writeln; writeln;
TextColor(15);
writeln('НАЖМИТЕ ЛЮБУЮ КЛАВИШУ');
ch:=readkey;
end;

procedure SetOut;
var i,j:integer;
begin
  {Данные для решения поставленной задачи}
  OutData;
  {L-характеристика транспортной сети}

```

```

OutL;
{Граф транспортной сети}
OutTS;
{Р-корреспонденция пассажиропотоков в час пик}
OutP;
{Т-время для пересадки}
OutT;
window(1,1,75,25);
end;

procedure start;
var
  code:word;
begin
repeat
  ClrScr;
  window(12,7,75,25);
  TextColor(10);
  writeln;
  writeln('                ПРОГРАММА ');
  writeln('        ВЫБОРА ОПТИМАЛЬНОЙ СХЕМЫ');
  writeln('    ГОРОДСКИХ АВТОБУСНЫХ МАРШРУТОВ');
  BarR(10,5,1,45,4);
  TextColor(15);
  Delay(2000);
  NLK;
  window(1,1,70,25);
  ClrScr;
  Setzn;
  SetOut;
  Rasch;
  ch:=readkey;
  ClrScr;
  BarR(10,20,8,30,4);
  GoToXY(27,10);
  TextColor(15);
  Write('ПОВТОРИТЬ? Д/Н');
  ch:=readkey;
until (ch<>'Д')and(ch<>'д')and(ch<>'I')and(ch<>'L')
end;

```

Окончание приложения 4

```
BEGIN  
  Color(15,0);  
  start;  
END.
```

Учебное издание

НАХОЖДЕНИЕ КРАТЧАЙШЕГО ПУТИ И ВЫБОР ОПТИМАЛЬНОЙ СХЕМЫ АВТОБУСНЫХ МАРШРУТОВ С ПОМОЩЬЮ ПЭВМ

Методические указания

Автор-составитель
Гудилов Сергей Владимирович

Редактор *Л. В. Ванюшина*
Компьютерная верстка *О. Г. Климантова*

Изд. лиц. ЛР № 020815 от 21.09.98.

Подписано в печать 20.03.2008. Бумага офсетная. Формат 60×84 1/16.

Гарнитура Times New Roman. Печать офсетная.

Усл. печ. л. 3,4. Уч.-изд. л. 3,8. Тираж 100 экз. Заказ №

Издательско-полиграфический центр

Новгородского государственного университета им. Ярослава Мудрого.

173003, Великий Новгород, ул. Б. Санкт-Петербургская, 41.

Отпечатано в ИПЦ НовГУ.

173003, Великий Новгород, ул. Б. Санкт-Петербургская, 41.