МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ДГТУ)

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ ПО ДИСЦИПЛИНЕ «Графы и конечные автоматы»

Предназначено для студентов первого курса заочной формы обучения по направлениям подготовки 09.03.02 «Информационные системы и технологии», 09.03.03 «Прикладная информатика»

г. Ростов-на-Дону

2022

В пособии приведены варианты заданий контрольной работы для студентов заочной формы обучения по темам, соответствующие базовому уровню изучения дисциплины «Графы и конечные автоматы», сформулированы методические рекомендации по их выполнению. Рассмотрены также образцы решения всех заданий и краткие теоретические сведения, используемые в этих решениях.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Учебно-методическое пособие содержит индивидуальные задания контрольной работы по дисциплине «Графы и конечные автоматы», изучаемой на втором курсе студентами заочной формы обучения. Тематика заданий контрольной работы охватывает следующие темы дисциплины: матричные характеристики графов, пути и остовы минимального веса, формальные грамматики и языки, конечные автоматы распознаватели и преобразователи.

Каждый вариант контрольной работы включает 6 заданий. Правило выбора варианта сформулировано в соответствии с номером зачётной книжки студента (по сумме двух последних цифр зачетки). Это правило приведено перед заданиями контрольной работы.

В пособии представлены основные теоретическими положения и понятия (конспект лекций), соответствующие базовому уровню изучения дисциплины по темам, включенным в задания контрольной работы, подробное решение и оформление всех заданий контрольной работы.

Приведен список теоретических вопросов для подготовки к экзамену по дисциплине «Графы и конечные автоматы», учебная литература, рекомендуемая для подготовки к экзамену, а также для более полного изучения материала.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ И ЗАДАНИЯ ДЛЯ КОНТРОЛЬНОЙ РАБОТЫ

Уважаемые студенты!

Для качественного освоения учебного материала дисциплины «Графы и конечные автоматы» и успешного выполнения заданий контрольной работы Вам необходимо, прежде всего, изучить теоретический материал, затем рассмотреть и проанализировать примеры решения практических задач. В этом Вам помогут разобраться указанные в нашем пособии информационные источники, а также приведенный в пособии краткий конспект лекций.

После этого вы можете приступать к выполнению заданий своего варианта. Будьте внимательны к определению номера вашего варианта.

При самостоятельном решении и оформлении заданий вашего варианта Вам будет полезно использовать рассмотренный в пособии образец выполнения и оформления всех заданий контрольной работы.

В пособии также приведены вопросы для подготовки к экзамену.

Варианты заданий контрольной работы

Правило выбора номера варианта контрольной работы: номером варианта служит число N, равное сумме двух последних цифр зачетной книжки студента, если это не ноль. Если же две последние цифры номера зачетной книжки студента нули, то следует полагать N=19.

Задание 1

Дан орграф с n вершинами и m дугами. U– множество дуг этого графа. Выполнить:

1)Изобразить граф. Определить, является ли граф планарным?

2)Найти матрицу смежности A графа, степени исхода и захода для вершин, проверить теорему Эйлера для степеней вершин графа.

3) Найти степени матрицы смежности A^2, A^3. С помощью степеней найти общее число путей из p-й и q-й вершин во все остальные с числом дуг не более трех.

4) Найти матрицу инцидентности B графа.

Варианты

N	данные варианта	N	данные варианта
1	n=5,m=8,p=1,q=5 U={(1,5),(5,1),(1,3), (3,5),(3,4),(4,5),(2,2),(2,4)}	11	n=6,m=8,p=2,q=4 U={(1,2),(2,5),(1,5), (2,3),(5,3),(5,4),(3,4),(6,6)}
2	n=6,m=7,p=2,q=6 U={(4,4),(2,6),(2,3), (3,6),(3,5),(5,1),(1,6)}	12	n=5,m=7,p=1,q=5 U={(1,2),(4,4),(1,3),(2,3), (3,5),(3,4),(5,4)}
3	n=6,m=8,p=3,q=2 U={(1,1),(3,1),(1,2), (1,3),(3,6),(6,2),(5,4),(4,4)}	13	n=5,m=6,p=1,q=3 U={(1,2),(2,3),(1,3), (3,4),(4,1),(5,5)}
4	n=5,m=7,p=1,q=4 U={(1,5),(1,2),(2,5), (3,5),(2,3),(3,4),(5,4)}	14	n=6,m=8,p=4,q=3 U={(1,5),(1,2),(2,3), (3,4),(4,2),(4,1),(4,5),(6,6)}
5	n=6,m=8,p=1,q=5 U={(1,2),(1,6),(2,6), (2,3),(6,5),(3,5),(3,4),(4,5)}	15	n=5,m=7,p=3,q=1 U={(1,2),(1,4),(4,5), (5,1),(2,3),(3,4),(3,1)}
6	n=6,m=8,p=1,q=4 U={(1,2),(6,1),(2,3),	16	n=5,m=7,p=2,q=5 U={(4,4),(2,5),(2,3),

	(6,3),(1,4),(3,4),(4,5),(5,6)}		(3,5),(3,3),(5,1),(1,5)}
7	n=6,m=8,p=3,q=1 U={(1,2),(1,4),(4,5),(6,6), (5,1),(2,3),(3,4),(3,1)}	17	n=7,m=9,p=3,q=5 U={(1,2),(1,4),(4,5),(2,6), (5,1),(2,3),(3,4),(3,1),(7,7)}
8	n=7,m=9,p=4,q=3 U={(1,5),(1,2),(2,3),(7,7), (3,4),(4,2),(4,1),(4,5),(6,7)}	18	n=5,m=7,p=4,q=3 U={(1,5),(1,2),(2,3), (3,4),(4,2),(4,1),(4,5)}
9	n=5,m=6,p=1,q=4 U={(1,2),(2,3),(1,5), (3,5),(3,4),(5,4)}	19	n=5,m=6,p=1,q=5 U={(1,2),(1,3),(2,3), (3,5),(3,4),(5,4)}
10	n=6,m=8,p=3,q=2 U={(1,1),(4,1),(1,2), (1,4),(4,5),(5,2),(6,3),(3,3)}	20	n=5,m=7,p=2,q=4 U={(1,2),(2,5),(1,5), (2,3),(5,3),(5,4),(3,4)}

Задание 2

Имеется склад (вершина 1) и шесть потребителей (вершины 2-7). Известны стоимости перевозок между складом и отдельными потребителям (матрица стоимостей). Элемент матрицы стоимости равен ∞, если соответствующие пункты (вершины) не связаны дорогой. Изобразить граф, указывая веса дуг. Найти оптимальные планы доставки грузов со склада всем потребителям (вершины 2-7), при которых стоимости перевозок будут минимальными. Пути минимального веса из 1 вершины во все остальные записать в виде последовательности вершин, указать также веса этих путей.

Варианты

1) $\begin{pmatrix} \infty & 2 & 3 & 1 & 12 & 13 & \infty \\ \infty & \infty & 4 & 3 & 2 & 3 & \infty \\ \infty & \infty & \infty & 2 & 1 & 4 & \infty \\ \infty & 1 & \infty & \infty & 2 & 3 & \infty \\ \infty & \infty & 2 & \infty & \infty & 1 & 3 \\ \infty & 4 & \infty & 4 & \infty & \infty & 4 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$	**2)** $\begin{pmatrix} \infty & 12 & 2 & 4 & 1 & 11 & \infty \\ \infty & \infty & 2 & 4 & 2 & \infty & 3 \\ \infty & \infty & \infty & 4 & 2 & 5 & \infty \\ \infty & 3 & \infty & \infty & 1 & 3 & \infty \\ \infty & \infty & 2 & \infty & \infty & 6 & \infty \\ \infty & 5 & 2 & 1 & \infty & \infty & 4 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$	**3)** $\begin{pmatrix} \infty & 5 & 13 & 11 & 4 & 3 & \infty \\ \infty & \infty & \infty & \infty & 2 & 1 & \infty \\ \infty & \infty & \infty & 2 & 3 & 3 & 2 \\ \infty & 3 & \infty & \infty & 4 & 3 & 5 \\ \infty & \infty & 2 & \infty & \infty & 4 & \infty \\ \infty & 5 & 2 & 1 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$
4) $\begin{pmatrix} \infty & 3 & 1 & 2 & 10 & 11 & \infty \\ \infty & \infty & 4 & 3 & \infty & 3 & \infty \\ \infty & \infty & \infty & 3 & 1 & 4 & \infty \\ \infty & 1 & \infty & \infty & 2 & 3 & \infty \\ \infty & \infty & 2 & \infty & \infty & 4 & 5 \\ 2 & \infty & \infty & 3 & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$	**5)** $\begin{pmatrix} \infty & 10 & 3 & 5 & 2 & 13 & \infty \\ \infty & \infty & 2 & 2 & 4 & \infty & 5 \\ \infty & \infty & \infty & 5 & 2 & 3 & \infty \\ 4 & 3 & \infty & \infty & 1 & 3 & \infty \\ \infty & \infty & 2 & \infty & \infty & 4 & \infty \\ \infty & 6 & \infty & 1 & \infty & \infty & 2 \\ \infty & 1 & \infty & \infty & \infty & 1 & \infty \end{pmatrix}$	**6)** $\begin{pmatrix} \infty & 3 & 12 & 10 & 2 & 4 & \infty \\ \infty & \infty & 9 & \infty & 5 & 3 & \infty \\ \infty & \infty & \infty & 2 & 3 & 3 & 3 \\ \infty & 3 & \infty & \infty & 4 & 3 & 4 \\ \infty & \infty & 2 & \infty & \infty & 2 & \infty \\ \infty & 5 & 2 & 1 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$
7) $\begin{pmatrix} \infty & 8 & 3 & 5 & \infty & 9 & \infty \\ \infty & \infty & 2 & 1 & 3 & \infty & 5 \\ 2 & \infty & \infty & 1 & 3 & 2 & \infty \\ \infty & 3 & \infty & \infty & 1 & 3 & \infty \\ \infty & \infty & 2 & \infty & \infty & 4 & \infty \\ \infty & 5 & 2 & 1 & \infty & \infty & 3 \\ \infty & 3 & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$	**8)** $\begin{pmatrix} \infty & 3 & 15 & 16 & 2 & 7 & \infty \\ \infty & \infty & \infty & \infty & 6 & 1 & \infty \\ \infty & \infty & \infty & 2 & 3 & 3 & 2 \\ \infty & 3 & \infty & \infty & 6 & 3 & 5 \\ \infty & \infty & 2 & \infty & \infty & 1 & \infty \\ \infty & 5 & 2 & 1 & \infty & \infty & \infty \\ \infty & \infty & 1 & \infty & \infty & \infty & \infty \end{pmatrix}$	**9)** $\begin{pmatrix} \infty & 3 & 4 & 2 & 14 & 12 & \infty \\ \infty & \infty & 4 & 3 & 2 & 6 & \infty \\ \infty & \infty & \infty & 2 & 1 & 4 & \infty \\ \infty & 1 & \infty & \infty & 4 & 3 & \infty \\ 3 & \infty & 2 & \infty & \infty & 1 & 5 \\ \infty & 4 & \infty & 4 & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$

10) $\begin{pmatrix} \infty & 3 & 2 & 2 & 15 & 12 & \infty \\ 1 & \infty & 6 & 4 & 1 & 4 & \infty \\ \infty & \infty & \infty & 1 & 5 & 4 & \infty \\ \infty & 1 & \infty & \infty & 2 & 6 & \infty \\ \infty & \infty & 2 & \infty & \infty & 1 & 7 \\ \infty & 4 & \infty & 4 & \infty & \infty & 5 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$	**11)** $\begin{pmatrix} \infty & 16 & 5 & 6 & 3 & 17 & \infty \\ \infty & \infty & 5 & 7 & 2 & \infty & 6 \\ \infty & \infty & \infty & 8 & 4 & 9 & \infty \\ \infty & 1 & \infty & \infty & 5 & 8 & \infty \\ \infty & \infty & 2 & \infty & \infty & 2 & \infty \\ \infty & 2 & 3 & 1 & \infty & \infty & 5 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$	**12)** $\begin{pmatrix} \infty & 6 & 18 & 19 & 2 & 4 & \infty \\ \infty & \infty & \infty & \infty & 2 & 3 & \infty \\ \infty & \infty & \infty & 2 & 3 & 3 & 4 \\ \infty & 3 & \infty & \infty & 4 & 3 & 5 \\ 2 & \infty & 4 & \infty & \infty & 4 & \infty \\ \infty & 5 & 4 & 1 & \infty & \infty & \infty \\ 1 & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$
13) $\begin{pmatrix} \infty & 4 & 2 & 3 & 15 & 14 & \infty \\ \infty & \infty & 5 & 6 & \infty & 4 & \infty \\ 3 & \infty & \infty & 5 & 3 & 4 & \infty \\ \infty & 1 & \infty & \infty & 2 & 3 & \infty \\ \infty & \infty & 8 & \infty & \infty & 3 & 7 \\ 2 & \infty & \infty & 3 & \infty & \infty & 8 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$	**14)** $\begin{pmatrix} \infty & 18 & 5 & 8 & 4 & 17 & \infty \\ \infty & \infty & 7 & 6 & 5 & \infty & 6 \\ \infty & \infty & \infty & 8 & 6 & 4 & \infty \\ 7 & 3 & \infty & \infty & 6 & 8 & \infty \\ \infty & \infty & 2 & \infty & \infty & 1 & \infty \\ \infty & 6 & \infty & 3 & \infty & \infty & 7 \\ \infty & 1 & \infty & \infty & \infty & 2 & \infty \end{pmatrix}$	**15)** $\begin{pmatrix} \infty & 7 & 23 & 20 & 9 & 8 & \infty \\ \infty & \infty & 12 & \infty & 15 & 13 & \infty \\ \infty & \infty & \infty & 12 & 13 & 9 & 5 \\ \infty & 10 & \infty & \infty & 7 & 8 & 4 \\ \infty & \infty & 2 & \infty & \infty & 6 & \infty \\ \infty & 5 & 2 & 1 & \infty & \infty & 24 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$
16) $\begin{pmatrix} \infty & 9 & 2 & 4 & \infty & 10 & \infty \\ \infty & \infty & 1 & 2 & 4 & \infty & 6 \\ 2 & \infty & \infty & 2 & 3 & 6 & \infty \\ \infty & 3 & \infty & \infty & 3 & 2 & \infty \\ \infty & \infty & 2 & \infty & \infty & 4 & \infty \\ \infty & 5 & 2 & 4 & \infty & \infty & 3 \\ \infty & 3 & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$	**17)** $\begin{pmatrix} \infty & 2 & 14 & 13 & 4 & 6 & \infty \\ \infty & \infty & \infty & 2 & 4 & 3 & \infty \\ \infty & \infty & \infty & 2 & 5 & 3 & 3 \\ \infty & 3 & \infty & \infty & 2 & 3 & 4 \\ \infty & \infty & 6 & \infty & \infty & 1 & \infty \\ \infty & 5 & 2 & 1 & \infty & \infty & 7 \\ \infty & \infty & 1 & \infty & \infty & \infty & \infty \end{pmatrix}$	**18)** $\begin{pmatrix} \infty & 2 & 5 & 1 & 16 & 14 & \infty \\ \infty & \infty & 2 & 3 & 7 & 6 & \infty \\ \infty & \infty & \infty & 2 & 6 & 4 & \infty \\ \infty & 1 & \infty & \infty & 8 & 2 & \infty \\ 3 & \infty & 2 & \infty & \infty & 1 & 6 \\ \infty & 4 & \infty & 4 & \infty & \infty & 7 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$
19) $\begin{pmatrix} \infty & 8 & 1 & 2 & 10 & 15 & \infty \\ \infty & \infty & 8 & 3 & \infty & 3 & \infty \\ \infty & \infty & \infty & 3 & 1 & 4 & \infty \\ \infty & 1 & \infty & \infty & 2 & 3 & \infty \\ \infty & \infty & 2 & \infty & \infty & 4 & 5 \\ 2 & \infty & \infty & 3 & \infty & \infty & 5 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$	**20)** $\begin{pmatrix} \infty & 12 & 3 & 5 & 2 & 20 & \infty \\ \infty & \infty & 5 & 2 & 4 & \infty & 7 \\ \infty & \infty & \infty & 5 & 2 & 3 & \infty \\ 4 & 3 & \infty & \infty & 1 & 3 & \infty \\ \infty & \infty & 2 & \infty & \infty & 4 & \infty \\ \infty & 6 & \infty & 1 & \infty & \infty & 4 \\ \infty & 1 & \infty & \infty & \infty & 1 & \infty \end{pmatrix}$	**21)** $\begin{pmatrix} \infty & 4 & 14 & 10 & 3 & 4 & \infty \\ \infty & \infty & 8 & \infty & 5 & 3 & \infty \\ \infty & \infty & \infty & 2 & 3 & 3 & 3 \\ \infty & 3 & \infty & \infty & 4 & 3 & 2 \\ \infty & \infty & 2 & \infty & \infty & 2 & \infty \\ \infty & 5 & 2 & 1 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$

Задание 3

Для неориентированного графа с множеством вершин {1,2,3,4,5}, множеством ребер U={I,II,III,IV,V,VI,VII,VIII} и множеством весов ребер W={w_1,w_2,w_3,w_4,w_5,w_6,w_7,w_8 } выполнить:

1) Изобразить граф, указывая номера вершин и веса ребер.

2) Найти остов минимального веса и вес остова, при этом указать последовательность добавления ребер.

Варианты

1	U={{1,2},{1,3},{1,4}, {1,5},{2,3},{2,4},{3,4}, {4,5}} W={2,3,3,4,1,4,2,3}	11	U={{2,3},{2,1},{2,4}, {2,5},{1,3},{1,4},{1,5}, {4,5}} W={7,7,5,8,6,5,8,7}
2	U={{1,2},{1,4},{5,1}, {2,5},{2,3},{2,4},{5,3}, {3,4}} W={9,2,5,1,6,4,8,3}	12	U={{1,3},{1,4},{2,4}, {2,3},{3,5},{1,2},{4,3}, {2,5}} W={6,4,3,2,3,1,5,5}

3	U={{1,3},{4,1},{1,2}, {2,5},{4,2},{4,3},{5,3}, {5,4}} W={3,7,5,1,6,1,8,6}	13	U={{5,2},{3,1},{2,4}, {3,2},{3,4},{1,4}, {1,5},{4,5}} W={2,3,9,5,1,4,2,1}
4	U={{1,3},{1,4},{1,2}, {2,5},{4,2},{3,4},{5,3}, {5,4}} W={1,5,2,7,4,1,8,9}	14	U={{3,4},{5,3},{1,3},{1,4}, {1,2},{2,5},{4,2}, {5,4}} W={2,1,5,4,6,2,9,10}
5	U={{1,2},{1,5},{5,3}, {1,3},{4,5},{2,4},{3,4}, {3,2}} W={6,2,1,8,9,4,2,1}	15	U={{1,3},{1,4},{2,4}, {2,3},{3,5},{1,2},{1,5}, {2,5}} W={8,4,9,2,1,5,8,6}
6	U={{2,4},{2,5},{3,4},{1,4}, {2,3},{2,1},{1,5},{4,5}} W={7,4,3,2,7,5,1,1}	16	U={{2,4},{2,5},{2,1},{1,5}, {3,4},{1,4},{2,3},{4,5}} W={1,2,5,2,9,5,3,4}
7	U={{2,4},{2,3},{3,5},{1,2}, {1,5},{1,3},{1,4},{2,5}} W={2,4,2,1,7,5,8,1}	17	U={{1,5},{4,5}{2,3},{2,1}, {2,4},{2,5},{3,4},{1,4}} W={3,5,1,2,9,4,2,3}
8	U={{1,3},{1,2},{4,3}, {1,4},{2,4},{2,3},{3,5}, {2,5}} W={8,4,3,6,3,1,7,2}	18	U={{2,3},{2,1},{2,4}, {2,5},{1,3},{1,4},{1,5}, {4,5}} W={3,7,6,1,6,5,8,2}
9	U={{1,3},{1,4},{1,2},{2,5}, {4,2},{3,4},{5,3},{5,4}} W={5,7,5,1,3,7,1,1}	19	U={{2,4},{2,5},{3,4},{1,4}, {2,3},{2,1},{1,5},{4,5}} W={3,5,8,2,7,5,8,6}
10	U={{1,2},{1,5},{5,2}, {1,3},{4,5},{2,4},{3,4}, {3,2}} W={1,2,1,7,1,4,2,1}	20	U={{2,3},{2,1},{2,4}, {2,5},{3,4},{1,4},{1,5}, {4,5}} W={2,4,1,2,3,5,8,2}

Задание 4

Для грамматики Г, заданной терминальным алфавитом V, нетерминальным алфавитом W, начальным символом (аксиомой) I, символом пустой цепочки $, множеством правил R выполнить:

1) Построить вывод слов формального языка L(Г), порождаемый этой грамматикой. Записать этот язык в виде общего выражения.

2)Перечислить первые 6 слов в лексикографическом (алфавитном) порядке.

Варианты

1	V={a,b},W={I,A} R={I→abA, A→aaAb, A→$}	11	V={0,3},W={I,A} R={I→0A33, A→0A03, A→$}
2	V={a,b,c},W={I,B} R={I→cBb, B→bBc, B→a, B→$}	12	V={a,b,c},W={I,A} R={I→aAbb, A→bbAa, A→c }
3	V={0,1},W={I,B} R={I→0B11, B→0B111, B→$}	13	V={0,2,3},W={I,B} R={I→0B2, B→2B0, B→3 }
4	V={0,1,2},W={I,A} R={I→00A11, A→00A1, A→22}	14	V={0,1},W={I,C} R={I→00C11, C→0C11, C→0}
5	V={0,1,3},W={I,A} R={I→0A1, A→1A01, A→3 }	15	V={0,1,5},W={I,B} R={I→00B1, B→1B00, B→5 }
6	V={0,1},W={I,B} R={I→00B11, B→0B11, B→$}	16	V={a,b,c},W={I,A} R={I→aaAbb, A→aAb, A→cc}

7	V={0,1,2},W={I,B} R={I→00B1, B→1B00, B→2}	17	V={a,b,c},W={I,B} R={I→aBb, B→bbBa, B→c }
8	V={0,1,4},W={I,A} R={I→00A1, A→00A1, A→44 }	18	V={0,1,3},W={I,A} R={I→1A00, A→00A1, A→33}
9	V={0,2},W={I,A} R={I→0A22, A→0A22, A→$}	19	V={m,b,n},W={I,C} R={I→mCb, C→bCm, C→n }
10	V={0,1},W={I,B} R={I→01B1, B→10I0, B→$}	20	V={1,2},W={I,A} R={I→1A2, A→11A2, A→$}

Задание 5

По таблице переходов конечного автомата-распознавателя (Q-алфавит состояний автомата, V-входной алфавит, начальное состояние - 1, заключительные состояния обведены в рамку выполнить:

1) Построить граф переходов автомата.

2) По графу переходов выписать в лексикографическом порядке первые 8 слов языка, распознаваемого автоматом.

Варианты

1	Q\V	a	b	c	**11**	Q\V	a	b	c
	1	1,2	–	–		1	–	–	2
	$\boxed{2}$	–	1	3		$\boxed{2}$	3,2	–	
	$\boxed{3}$	2	–	–		$\boxed{3}$	–	1	–
2	Q\V	a	b	c	**12**	Q\V	a	b	c
	$\boxed{1}$	–	2	–		$\boxed{1}$	3	–	1
	2	3	–	1		2	–	2	–
	$\boxed{3}$	2,3	–	–		$\boxed{3}$	–	–	2,1
3	Q\V	a	b	c	**13**	Q\V	a	b	c
	$\boxed{1}$	–	–	2,1		$\boxed{1}$	–	2	3
	$\boxed{2}$	–	3	–		$\boxed{2}$	–	–	–
	3	1	–	–		3	2	3	–
4	Q\V	a	b	c	**14**	Q\V	a	b	c
	1	–	3	–		1	–	3	–
	$\boxed{2}$	1	–	2		$\boxed{2}$	–	3,2	–
	$\boxed{3}$	2,1	–	–		$\boxed{3}$	2	–	1
5	Q\V	a	b	c	**15**	Q\V	a	b	c
	$\boxed{1}$	2,1	–	3		$\boxed{1}$	–	–	2,1
	2	–	1	–		2	3	–	–
	$\boxed{3}$	–	–	1		$\boxed{3}$	–	2	–
6	Q\V	a	b	c	**16**	Q\V	a	b	c
	$\boxed{1}$	3	–	–		$\boxed{1}$	–	–	3,1
	$\boxed{2}$	–	–	2		$\boxed{2}$	2	3	–
	3	1	2,1	–		3	–	–	1
7	Q\V	a	b	c	**17**	Q\V	a	b	c
	1	2	–	–		$\boxed{1}$	–	2,1	–
	$\boxed{2}$	–	3,1	–		2	–	–	1,3
	$\boxed{3}$	2	–	3		$\boxed{3}$	2	–	–
8	Q\V	a	b	c	**18**	Q\V	a	b	c
	$\boxed{1}$	–	3	–		$\boxed{1}$	–	–	2,1
	2	2	–	3		2	3	–	–
	$\boxed{3}$	2,1	–	–		$\boxed{3}$	–	2	–
9	Q\V	a	b	c	**19**	Q\V	a	b	c

	$\boxed{1}$	–	–	2,1		$\boxed{1}$	–	–	3,1
	$\boxed{2}$	–	3	–		$\boxed{2}$	2	3	–
	3	1	–	–		3	–	–	1
10	Q\V	a	b	c	**20**	Q\V	a	b	c
	$\boxed{1}$	–	2	-		$\boxed{1}$	–	2,1	–
	2	3,2	–	1		2	–	–	1,3
	$\boxed{3}$	–	2	–		$\boxed{3}$	2	–	–

Задание 6

Для автомата-преобразователя S с входным алфавитом $X=\{a,b,c\}$, выходным алфавитом $Y=\{0,1\}$, алфавитом состояний $Q=\{1,2,3\}$, заданного таблицей переходов-выходов (на пересечении строки и столбца в таблице стоит пара *«состояние, выходной символ»* выполнить:

1) нарисовать граф переходов, записывая в его вершинах номера состояний, а вдоль дуг пару *«входной/выходной символ»* .
2) по входному слову x найти выходное слово y, если начальное состояние $q_1=1$., заключительным может быть любое состояние.Таблица переходов-выходов

Q \ X	a	b	c
1	m,0	n,1	k,1
2	n,1	k,0	k,1
3	m,0	n,1	n,0

Варианты

1	x=aabbb, m=1, n=3, k=2	11	x=babba, m=2, n=3, k=1
2	x=abbba, m=2, n=3, k=1	12	x=ababa, m=3, n=2, k=1
3	x=aabab, m=3, n=1, k=2	13	x=aaabb, m=2, n=1, k=3
4	x=ababa, m=3, n=2, k=1	14	x=aaabb, m=3, n=1, k=2
5	x=babab, m=2, n=1, k=3	15	x=aabbb, m=1, n=2, k=3
6	x=ababb, m=3, n=1, k=2	16	x=bbaaa, m=1, n=2, k=3
7	x=aabab, m=3, n=2, k=1	17	x=abbab, m=1, n=3, k=2
8	x=bbbaa, m=1, n=2, k=3	18	x=abbaa, m=2, n=3, k=1
9	x=babab, m=2, n=1, k=3	19	x=aabab, m=3, n=1, k=2
10	x=ababa, m=3, n=2, k=1	20	x=abbba, m=1, n=2, k=3

ОБРАЗЕЦ ВЫПОЛНЕНИЯ И ОФОРМЛЕНИЯ ЗАДАНИЙ ВАРИАНТА КОНТРОЛЬНОЙ РАБОТЫ

Задание 1

Дан орграф с n вершинами и m дугами. U- множество дуг этого графа. Выполнить:

1)Изобразить граф. Является ли граф планарным?

2)Найти матрицу смежности A графа, степени исхода и захода для вершин, проверить теорему Эйлера для степеней вершин графа.

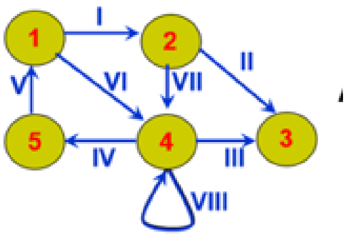
3) Найти степени матрицы смежности A^2, A^3. С помощью степеней найти общее число путей из p-й и q-й вершин во все остальные с числом дуг не более трех.

4) Найти матрицу инцидентности B графа.

Исходные данные варианта: n=5,m=8,p=2,q=4, U={(1,2),(1,4),(2,3),(2,4),(4,3),(4,5),(4,4),(5,1)}

Выполнение задания 1

1) Изобразим граф. Граф является планарным, так как может быть изображен на плоскости так, чтобы его дуги не имели пересечений, как показано на рисунке.



$$A=\begin{array}{c}\begin{array}{ccccc}1&2&3&4&5\end{array}\\ \begin{pmatrix}0&1&0&1&0\\0&0&1&1&0\\0&0&0&0&0\\0&0&1&1&1\\1&0&0&0&0\end{pmatrix}\end{array}\begin{array}{c}\\1\\2\\3\\4\\5\end{array}$$

2) Найдем матрицу смежности A (это булева матрица 5 на 5). Будем заполнять ее по строкам, рассматривая для каждой вершины только исходящие дуги (им соответствуют единицы в соответствующих строках матрицы):

Убеждаемся в справедливости теоремы Эйлера о степенях вершин:

$$\sum_{x\in X}\deg_-(x)+\sum_{x\in X}\deg_+(x)=2|U|=2m$$

$(2+2+0+3+1)+(1+1+2+3+1)=16=2\cdot 8$ (сумма элементов по строкам и столбцам матрицы смежности равна удвоенному числу дуг).

3) Найдем степени матрицы смежности

$$A^2=\begin{pmatrix}0&1&0&1&0\\0&0&1&1&0\\0&0&0&0&0\\0&0&1&1&1\\1&0&0&0&0\end{pmatrix}\begin{pmatrix}0&1&0&1&0\\0&0&1&1&0\\0&0&0&0&0\\0&0&1&1&1\\1&0&0&0&0\end{pmatrix}=\begin{pmatrix}0&0&2&2&1\\0&0&1&1&1\\0&0&0&0&0\\1&0&1&1&1\\0&1&0&1&0\end{pmatrix}$$

$$A^3=\begin{pmatrix}0&1&0&1&0\\0&0&1&1&0\\0&0&0&0&0\\0&0&1&1&1\\1&0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&2&2&1\\0&0&1&1&1\\0&0&0&0&0\\1&0&1&1&1\\0&1&0&1&0\end{pmatrix}=\begin{pmatrix}1&0&2&2&2\\1&0&1&1&1\\0&0&0&0&0\\1&1&1&2&1\\0&0&2&2&1\end{pmatrix}$$

По условию p=2,q=4, поэтому требуется определить количество путей из 2-ой и 4-ой вершин во все остальные. Для этого нужно рассмотреть вторую и четвертую строки матрицы $A+A^2+A^3$.

$$A+A^2+A^3=\begin{pmatrix}0&1&0&1&0\\0&0&1&1&0\\0&0&0&0&0\\0&0&1&1&1\\1&0&0&0&0\end{pmatrix}+\begin{pmatrix}0&0&2&2&1\\0&0&1&1&1\\0&0&0&0&0\\1&0&1&1&1\\0&1&0&1&0\end{pmatrix}+\begin{pmatrix}1&0&2&2&2\\1&0&1&1&1\\0&0&0&0&0\\1&1&1&2&1\\0&0&2&2&1\end{pmatrix}=\begin{pmatrix}1&1&4&5&3\\1&0&3&3&2\\0&0&0&0&0\\2&1&3&4&3\\1&1&2&3&1\end{pmatrix}.$$

Из второй вершины в вершины 1,2,3,4,5 количество путей с числом дуг не более 3 равно соответственно 1,0,3,3,2.

Из четвертой вершины в вершины 1,2,3,4,5 количество путей с числом дуг не более 3 равно соответственно 2,1,3,4,3.

Справедливость этих утверждений проверяем непосредственно по чертежу графа.

4) Найдем матрицу инцидентности графа B. Размерность матрицы будет 5 на 8. Заполним матрицу по столбцам, записывая в его строках, соответствующих инцидентным вершинам, -1 для исхода и 1 для захода:

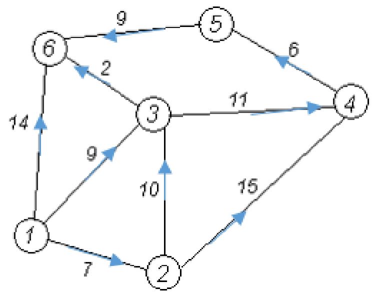
$$
\begin{array}{c}
\begin{array}{cccccccc} \mathrm{I} & \mathrm{II} & \mathrm{III} & \mathrm{IV} & \mathrm{V} & \mathrm{VI} & \mathrm{VII} & \mathrm{VIII} \end{array}\\
\mathrm{B} = \begin{pmatrix}
-1 & & & & 1 & -1 & & \\
1 & -1 & & & & & -1 & \\
 & 1 & 1 & -1 & & & & \\
 & & -1 & 1 & & 1 & 1 & \pm 1 \\
 & & & & -1 & & &
\end{pmatrix}
\end{array}
$$

Задание 2

Имеется склад (вершина 1) и пять потребителей (вершины 2-6). Известны стоимости перевозок между складом и отдельными потребителям (матрица стоимостей). Элемент матрицы стоимости равен ∞, если соответствующие пункты (вершины) не связаны дорогой. Изобразить граф, указывая веса дуг. Найти оптимальные планы доставки грузов со склада всем потребителям (вершины 2-6), при которых стоимости перевозок будут минимальными. Пути минимального веса из 1 вершины во все остальные записать в виде последовательности вершин, указать также веса этих путей.

Исходные данные варианта: дана матрица весов:

$$
\begin{pmatrix}
\infty & 7 & 9 & \infty & \infty & 14 \\
\infty & \infty & 3 & 15 & \infty & \infty \\
\infty & \infty & \infty & 11 & \infty & 2 \\
\infty & \infty & \infty & \infty & 6 & \infty \\
\infty & \infty & \infty & \infty & \infty & 9 \\
\infty & \infty & \infty & \infty & \infty & \infty
\end{pmatrix}
$$



Выполнение задания 2

Изобразим граф, соответствующий матрице стоимостей.

Для определения путей минимального веса из 1 первой вершины во все остальные воспользуемся алгоритмом Дейкстры. Суть алгоритма в следующем: в процессе итерационного поиска минимального маршрута всем вершинам графа приписываются метки (числа), рассчитываемые по определенным формулам. Метки подразделяются на постоянные (они подчеркиваются или выделяются цветом) и временные. Если какая-либо вершина получила постоянную метку, значит, для нее найден путь минимального веса, а значение постоянной метки равно минимальному весу пути в эту вершину из первой вершины. На каждой итерации только одна вершина получает постоянную метку, и эта вершина используется в качестве начальной на следующей итерации.

Процесс вычисления меток для наглядности представим в виде таблицы, которая заполняется по строкам, соответствующим итерациям. На каждой итерации линиями показываем соседей начальной вершины, метка которой постоянна.

В столбцах таблицы указываем метки вершин, которые они имеют в процессе работы алгоритма. Так в первом столбце указаны метки вершины 1, во втором метки второй вершины и т.д.

Для восстановления впоследствии кратчайшего маршрута рядом с каждой подчеркнутой меткой в скобках указываем номер начальной вершины, метка которой постоянна. Если таких вершин несколько, выбирается одна любая из них. Заметим, что путь минимального веса может быть определен не единственным способом, а вес минимального пути находится однозначно.

Нулевая итерация. Вершина 1– начальная, ей приписывается постоянная метка 0 (она подчеркнута), а всем остальным вершинам – временные метки ∞ .

Первая итерация. Минимальную метку 0 на нулевой итерации имеет вершина 1. Это начальная вершина для следующей итерации. Её достижимыми соседями являются вершины 2, 3 и 6. Первый по очереди сосед вершины 1 – вершина 2. Длина маршрута в неё через вершину 1 равна сумме значения метки начальной вершины 1 и длины ребра, идущего из 1-й в 2-ю, то есть 0 + 7 = 7. Это меньше текущей метки 2-ой вершины, равной ∞, поэтому новая метка 2-й вершины равна 7. Аналогичную операцию проделываем с двумя другими соседями 1-й вершины — 3-й и 6-й. Полученные соответственно метки

9 и 14 заносим в столбцы 3 и 6 таблицы. В скобках указываем 1- номер начальной вершины на этом шаге. Из полученных значений выбираем минимальное 7, это теперь постоянная метка, подчеркиваем ее и берем вершину 2 за начальную на следующей итерации.

Вторая итерация.

Достижимыми соседями начальной вершины 2 являются вершины 3 и 4. Первый по порядку сосед вершины 2 – вершина 3. Если идти в неё через 2, то длина такого маршрута будет равна 17 (7 + 10 = 17). Но текущая метка 3-ей вершины равна 9, а это меньше 17, поэтому метка этой вершины не меняется. Ещё один сосед вершины 2 –вершина 4. Если идти в неё через 2-ю, то длина такого пути будет равна метке 2-й вершины и расстояния между вершинами 2 и 4, то есть 22 (7 + 15 = 22). Поскольку 22< ∞ , устанавливаем метку вершины 4 равной 22.

метка вершины шаг k	$\lambda_1^{(к)}$	$\lambda_2^{(к)}$	$\lambda_3^{(к)}$	$\lambda_4^{(к)}$	$\lambda_5^{(к)}$	$\lambda_6^{(к)}$
Нулевая итерация (k=0)	0	∞	∞	∞	∞	∞
Первая итерация (k=1)		7(1)	9(1)	∞	∞	14(1)
Вторая итерация (k=2)			9(1)	22(2)	∞	14(1)
Третья итерация				20(3)	∞	11(3)
Четвертая итерация				20(3)	20(6)	
Пятая итерация				20(3)		

Третья итерация. Повторяем алгоритм, выбрав за начальную вершину 3. После ее обработки результаты занесем в таблицу и вершину 6 возьмем за начальную и т.д.

Результат работы алгоритма находим по таблице: кратчайший маршрут от вершины 1 до 2-й составляет 7, до 3-й — 9, до 4-й — 20, до 5-й — 20, до 6-й — 11.

Восстанавливаются эти кратчайшие маршруты по номерам вершин в скобках в таблице.

Например, восстановим путь из вершины 1 в вершину 5. Двигаемся из конечной вершины 5, в которую в минимальном пути ведет дуга из вершины 6. В вершину 6 попадаем из вершины 3, в 3 приходим из начальной вершины 1.Следовательно, искомый кратчайший маршрута: 1- 3- 6- 5 (вес пути равен 20).

путь	вес пути
1-2	7
1-3	9
1-3-6	11
1- 3- 6- 5	20
1-3-4	20

Задание 3

Для неориентированного графа с множеством вершин {1,2,3,4,5}, множеством ребер U={I,II,III,IV,V,VI,VII,VIII} и множеством весов ребер W={w_1,w_2,w_3,w_4,w_5,w_6,w_7,w_8 } выполнить:

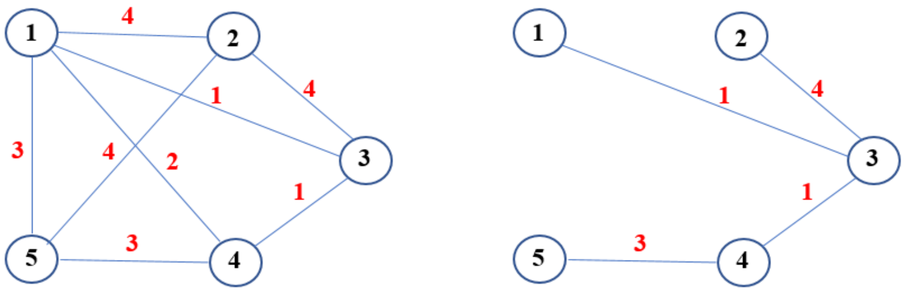
1) Изобразить граф, указывая номера вершин и веса ребер.

2) Найти остов минимального веса и вес остова, при этом указать последовательность добавления ребер.

Исходные данные варианта: U={{1,2},{1,4},{1,5}, {1,3},{2,3},{2,5},{3,4},{4,5}}, W={4,2,3,1,4,4,1,3}.

Выполнение задания 3

1) Изобразим неориентированный взвешенный граф



2) Найдем остов минимального веса по алгоритму Краскала. Так как $n=|X|=5$. то $|U_o|=|X|-1=4$ - число ребер в остове. Вначале строим пустой граф, содержащий только вершины исходного графа. Последовательно добавляем ребра возможно меньшего веса так, чтобы не образовывалось циклов.

Последовательность добавления ребер следующая: $\mathbf{U_0}$={{1,3}, {3,4}, {4,5}, {2,3}}. Ребро {1,4} не добавлено, так как его добавление привело бы к образованию цикла.

Заметим, что остов минимального веса может быть другим, например, вместо ребра {4,5} можно было добавить ребро {1,5}. Добавить сразу два ребра {4,5}, {1,5} также нельзя, так как образовался бы цикл. Вес минимального остова определяется однозначно, несмотря на неоднозначность построения остова.

Вес остова минимального веса: $W(U_0)$=1{3,1}+1{3,4}+3{4,5}+4{5,2}=9.

Задание 4

Для КС–грамматики Г, заданной терминальным алфавитом V, нетерминальным алфавитом W, начальным символом I, символом пустой цепочки $, множеством правил R выполнить:

1) Построить вывод слов формального языка L(Г), порождаемый этой грамматикой. Записать этот язык в виде общего выражения.

2)Перечислить первые 6 слов в лексикографическом порядке.

Исходные данные варианта: Г: V= {a, b}, W = {I, A}, R = {I $\to$ aA|$, A $\to$ bA|a}.

Выполнение задания 4

1) Выпишем последовательность (вывод) цепочек согласно правилам R, начиная с аксиомы I. Цепочки, содержащие нетерминалы, являются промежуточными. В рамочку обводим конечные цепочки, принадлежащие языку.

$$\mathrm{I} \Rightarrow \left[\begin{matrix}\mathrm{aA}\\ \boxed{\$}\end{matrix}\right. \Rightarrow \left[\begin{matrix}\mathrm{abA}\\ \boxed{aa}\end{matrix}\right. \Rightarrow \left[\begin{matrix}\mathrm{abbA}\\ \boxed{aba}\end{matrix}\right. \Rightarrow \dots \left[\begin{matrix}\mathrm{ab}^{n+1}\mathrm{A}\\ \boxed{ab^n a}\end{matrix}\right. , \dots ..$$

Получим язык в виде общего выражения: L(Г)={$}U{$ab^n a, n = 0,1,2 \dots ..$ }.

2) Перечислим в лексикографическом порядке первые шест слов языка: $\$, aa, aba, abba, abbba, abbbba.$

Задание 5

По таблице переходов конечного автомата-анализатора (Q-алфавит состояний, V- входной алфавит, начальное состояние - 1, заключительные состояния обведены в рамку. Выполнить:

1) Построить граф переходов автомата.

2) По графу переходов выписать в лексикографическом порядке первые 8 слов языка, распознаваемого автоматом.

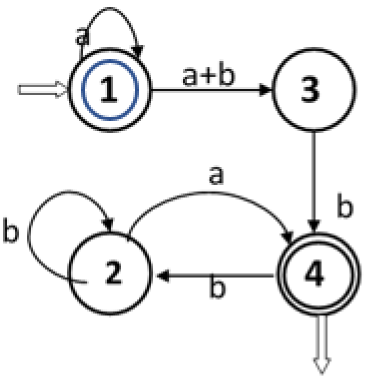
Исходные данные варианта:

Q\V	*a*	*b*
1	*3, 1*	*3*
2	*4*	*2*
3	-	*4*

4	-	2

Выполнение задания 5

1) Построим граф переходов автомата в виде размеченного графа. Вершины графа соответствуют состояниям, дуги – командам, записанным в таблице. Начальное и заключительное состояния помечены жирными стрелками. Выражение a+b заменяет кратные дуги, помеченные символами a и b.



2) Определим первые 8 слов языка и запишем их в лексикографическом порядке. Слово распознается автоматом и принадлежит языку, если существует путь из начального состояния в заключительное, на дугах которого последовательно записаны символы этого слова. Находим: $\{ab, bb, aab, abb, aaab, aabb, abba, bbba....\}$

Задание 6

Для автомата-преобразователя S с входным алфавитом $X = \{a, b, c\}$, выходным алфавитом $Y = \{0,1\}$, алфавитом состояний $Q = \{1,2,3\}$, заданного таблицей переходов-выходов (на пересечении строки и столбца в таблице стоит пара *«состояние, выходной символ»* выполнить:

2) нарисовать граф переходов, записывая в его вершинах номера состояний, а вдоль дуг пару *«входной/выходной символ»* .

2) по входному слову x найти выходное слово y, если начальное состояние $q_н = 1$, заключительным может быть любое состояние.

Таблица переходов-выходов

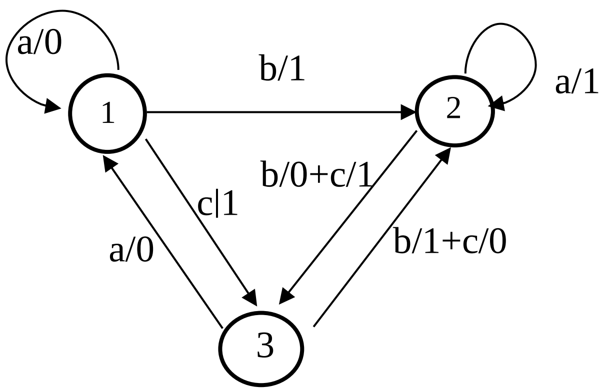
X / Q	a	b	c
1	m,0	n,1	k,1
2	n,1	k,0	k,1
3	m,0	n,1	n,0

Исходные данные варианта: x=ababc, m=1, n=2, k=3

Выполнение задания 6

1) Составим таблицу переходов-выходов для своего варианта и в соответствии с таблицей нарисуем граф переходов

X / Q	a	b	c
1	1,0	2,1	3,1
2	2,1	3,0	3,1
3	1,0	2,1	2,0



2) Найдем по входному слову выходное, используя команды таблицы или графа переходов $S(1,ababc)=0S(1,babc)=01S(2,abc)=011S(2,bc)=0110S(3,c)=01100S(2)$.

Таким образом, преобразованное слово 01100.

КОНСПЕКТ ЛЕКЦИЙ

Понятие графа, степеней вершин

Определение. Орграф G – это совокупность множества вершин и множества дуг: $G = \langle X, U \rangle$, причем множество дуг U задается в виде упорядоченных (x_i, x_j) пар вершин.

Определение. Неориентированный граф G – это совокупность множества вершин и множества ребер: $G = \langle X, U \rangle$, причем множество ребер U задается в виде неупорядоченных $\{x_i, x_j\}$ пар вершин.

Определение. Степенью вершины X называется число инцидентных ей дуг, для орграфа полустепенью исхода $\deg_-(x)$ называется число дуг, исходящих из вершины X, полустепенью захода $\deg_+(x)$ – число дуг, заходящих в нее[1], степень вершины в орграфе $\deg(x) = \deg_-(x) + \deg_+(x)$.

Теорема Эйлера о степенях вершин: для орграфа

$$\sum_{x \in X} \deg_-(x) + \sum_{x \in X} \deg_+(x) = 2|U| ; \qquad \sum_{x \in X} \deg_-(x) = \sum_{x \in X} \deg_+(x) = |U| .$$

Виды графов, планарность

На рис.1 изображены основные виды графов.

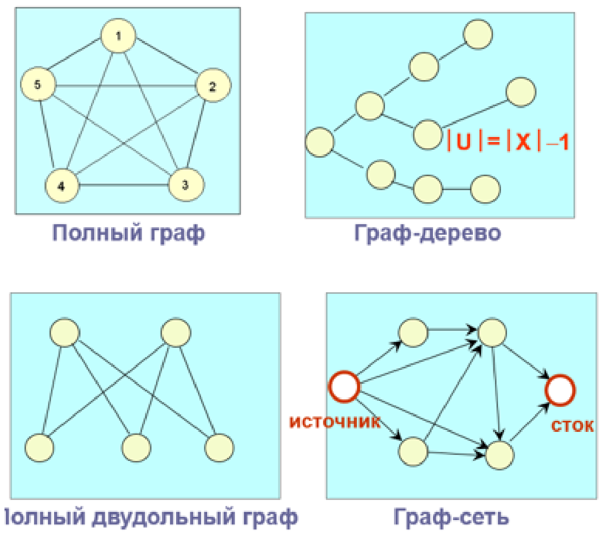


Рисунок 1

Определение. Графы, которые могут быть изображены на плоскости без пересечения их ребер в точках, отличных от вершин, называются плоскими или планарными.

Матрицы смежности и инцидентности

Матрицей смежности $A(G)$ n-вершинного орграфа без кратных дуг называется квадратная матрица порядка n, элементы которой определяются следующим:

---

[1] В учебной литературе могут встречаться другие обозначения степеней вершин, например, $\deg_-(x)$ или $\delta_-(x)$ обозначается число дуг, заходящих в вершину X, $\deg_+(x)$ и $\delta_+(x)$ – число дуг, исходящих из вершины X.

$$a_{ij} = \begin{cases} 1, \text{ если } (x_i, x_j) \in U; \\ 0, \text{ если } (x_i, x_j) \notin U. \end{cases}$$

Матрица смежности графа без кратных дуг является булевой или бинарной матрицей (матрицей, элементы которой 0 и 1). Дуги вида (x_i, x_i) соответствуют петлям.

Если орграф имеет кратные дуги, то в матрице смежности вместо 1 ставится число, равное кратности дуги.

Матрицу смежности удобнее заполнять по строкам. Для каждой вершины рассматриваются только исходящие дуги. В j-ом столбце i-й строки ставим «1», если если $(x_i, x_j) \in U$, в противном случае «0».

Пример. Найти матрицу смежности графа, изображенного на рис 2.

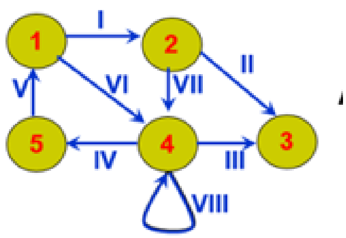


Рисунок 2

Решение.

$$A = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}$$

<u>Матрицей инцидентности (или инциденций)</u> $B(G)$ орграфа, имеющего n вершин и m дуг, называется матрица размерности $n \times m$, элементы которой определяются следующим:

$$b_{ij} = \begin{cases} +1, \text{ если вершина } x_i \text{ является концом дуги } u_j, \\ -1, \text{ если вершина } x_i \text{ является началом дуги } u_j, \\ 0, \text{ если вершина } x_i \text{ и дуга } u_j \text{ не инцидентны }, \\ \pm 1, \text{ если дуга } u_j \text{ является петлей в вершине } x_i. \end{cases}$$ [2]

Матрицу инцидентности удобнее заполнять по столбцам.

Задача. Найти матрицу инцидентности графа, изображенного на рис. 2.

Решение.

$$B = \begin{array}{c} \begin{array}{cccccccc} \mathrm{I} & \mathrm{II} & \mathrm{III} & \mathrm{IV} & \mathrm{V} & \mathrm{VI} & \mathrm{VII} & \mathrm{VIII} \end{array} \\ \begin{pmatrix} -1 & & & & 1 & -1 & & \\ 1 & -1 & & & & & -1 & \\ & 1 & 1 & -1 & & & & \\ & & -1 & 1 & & 1 & 1 & \pm 1 \\ & & & & -1 & & & \end{pmatrix} \end{array}$$

Пути, маршруты, связность

Путем на орграфе называют такую последовательность дуг, в которой начало каждой последующей дуги совпадает с концом предыдущей.

Путь задают в виде последовательности чередующихся вершин и дуг или просто перечислением вершин или дуг.

Путь называется простым, если никакая вершина в нем не встречается дважды. Если между какими-либо двумя вершинами существует путь, то между этими вершинами существует и простой путь.

Замкнутый путь называется контуром.

Маршрутом или цепью на неориентированном графе называется любая последовательность смежных ребер (ориентация не существенна). Цикл – замкнутый маршрут.

Элемент ak_{ij} матрицы A^k орграфа (графа) равен числу всех путей (маршрутов) , содержащих k дуг, из x_i в x_j, где $A-$ матрица смежности. При возведении в степень матрицы смежности используются арифметические (не логические) операции умножения и сложения.

Пример. Для орграфа задана матрица смежности. Найти число путей из первой вершины во все остальные с числом дуг не более двух, используя A^2.

$$A=\begin{pmatrix} 0 & \boxed{1} & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Решение. Перемножим А на А по правилу строка на столбец. Получим

$$A^2=\begin{pmatrix} 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}=\begin{pmatrix} 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

$$A^2=\begin{pmatrix} 0 & 0 & 0 & \boxed{1} & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Так как исходящие из первой вершины пути соответствуют ненулевым элементам первых строк матриц А и A^2, то искомое число путей равно двум.

Неориентированный граф называется связным, если любые его вершины соединены маршрутом. Числом связности графа называется число связных частей графа.

Орграф называется сильно связным, если из любой вершины существует путь в любую другую вершину. Если для любых двух вершин существует путь хотя бы в одном направлении, то орграф является односторонне связным.

Минимальные пути. Алгоритм Дейкстры

Матрицей весов (длин) дуг $W=[w_{ij}]$ называется матрица порядка n, для которой

$$w_{ij}=\begin{cases} w(v_i,v_j), & (v_i,v_j)\in U,\\ \infty & (v_i,v_j)\notin U. \end{cases}$$

Путь в нагруженном орграфе из вершины v_1 в вершину v_l, где $v_1 \neq v_l$, называется минимальным (кратчайшим) или путем наименьшего веса, если он имеет наименьшую сумму весов дуг, входящих в него.

Для неотрицательных весов w_{ij} путь минимального веса для n-вершинного графа из вершины v_1 в вершину v_n может включать не более $n-1$ дуг, так как этот путь – простой.

Алгоритм Дейкстры отыскания кратчайшего пути (маршрута) на графе

Обозначим меткой $\lambda_i^{(к)}$ – минимальный вес пути из начальной вершины в i-ую на к-ом шаге; содержащий не более, чем k, дуг. Процесс присваивания меток происходит итерационно.

- **Итерация (шаг) 0:** Выбираем начальную вершину. Помечаем начальную вершину меткой $\lambda_j^{(0)}=0$, подчеркиваем ее, эта вершина пройдена, остальные помечаем символом $\lambda_i^{(0)}=\propto$, $i\neq j$. Заполняем нулевую строку таблицы.
- **Итерация (шаг) k** (k=1,2,…n-1 ,n вершин в графе):
  Помечаем вершины всех соседей начальной j-ой вершины (которые еще не помечены ранее) метками по правилу: к старой метке соседа прибавляем вес дуги из начальной вершины к этому соседу; если полученное значение окажется меньше старой строй метки соседа, то приписываем эту новую метку соседу; если полученное значение не меньше, то оставляем старую метку. Этот алгоритм выражается формулой

$$\boxed{\lambda_i^{(к)} = \min(\lambda_i^{(к-1)}, \lambda_j^{(к-1)} + w_{ji}),}$$

где w_{ji}– вес дуги (j,i), $\lambda_j^{(к-1)}$ - метка начальной вершины, $\lambda_i^{(к-1)}$ - метки ее соседей (смежных вершин) на предыдущем шаге.

Метки вершин, не являющихся соседями, оставляем без изменений; заполняем k-ю строку таблицы.

Выбираем из меток соседей минимальную и делаем соответствующую ей вершину начальной, j-ой (подчеркнутой, пройденной); метка этой вершины больше не будет меняться, она равна минимальному весу пути в эту вершину.

Процесс продолжаем до тех пор, пока все вершины не будут пройдены.

Графы–деревья. Алгоритм Краскала

Определение. Деревом называется связный граф без циклов (рис. 4).

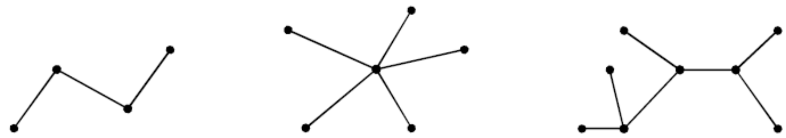


Рисунок 3

Свойства деревьев

1. В дереве число ребер на единицу меньше числа вершин $|U|=|X|-1$. Деревья имеют наименьшее количество ребер среди всевозможных связных графов, которые можно образовать по данному множеству вершин.
2. Любое ребро в дереве является мостом, т.е. ребром, удаление которого приводит к нарушению связности графа.
3. Соединение в дереве двух не смежных вершин одним ребром приводит к образованию ровно одного простого цикла.

Определение. Остовом связного графа G называется его подграф, обозначим G_o, являющийся деревом, вершинами которого являются все вершины данного графа G.

Пусть G– конечный связный неориентированный граф с *n* вершинами. Известны веса его ребер. Требуется найти соединение вершин, сумма весов ребер которого минимальна.

Искомое соединение является остовом минимального веса.

Для решения этой задачи используются алгоритмы Краскала, Прима и др.

Алгоритм Краскала нахождения остова минимального веса

– **0 итерация.** Найти количество ребер в остове по формуле $|U_o| = |X| - 1 = n - 1$.
  Построить пустой граф O_n, не содержащий ребер и включающий все вершины исходного графа G. Полученный граф является подграфом исходного графа.
– **К-я итерация, k=1,2,…n-1.** Присоединить к нему какое–либо ребро с возможным наименьшим весом из графа G так, чтобы при этом не образовалось цикла (если таких ребер несколько, то выбрать любое). Процесс продолжать, пока количество ребер в остове не станет равным n-1.

Рассчитать сумму весов ребер, входящих в остов.

Замечание. Остов минимального может быть не единственный.

Пример. Заданы множество ребер неориентированного графа или множество весов его ребер (указано в скобках).
U={I{1,2},II{1,5},III{1,8},IV{2,3},V{2,4},VI{2,5},VII{3,4},VIII{3,6},IX{3,7},X{3,8},XI{4,5},XII{4,7}, XIII{4,8},XIV{5,8},XV{6,7},XVI{7,8}}.
W={I(1),II(3),III(2),IV(3),V(2),VI(5),VII(3),VIII(1),IX(5),X(3),XI(1),XII(4),XIII(1),XIV(1), XV(4), XVI(4)}. Требуется найти остов минимального веса.

Решение. Так как $|X| = 8$. то $|U_o| = |X| - 1 = 7$- число ребер в остове. Вначале строим пустой граф. Последовательно добавляем ребра, чтобы не образовывалось циклов, возможно меньшего веса.
На рис. 5 показана последовательность добавления ребер.

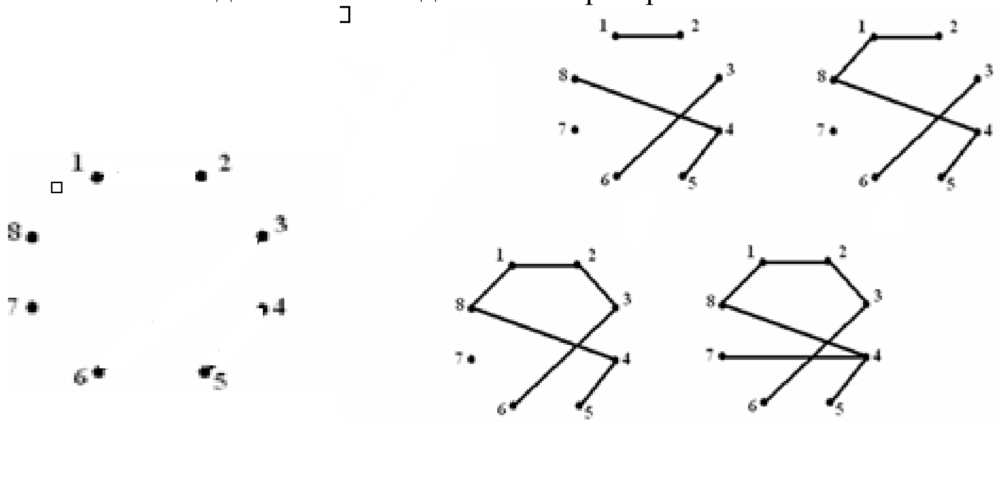


Рисунок 4

U₀={{1,2}, {4,5}, {3,6}, {4,8}, {1,8}, {2,3}, {4,7}}

Вес полученного дерева W(U_0)=1{1,2}+1{4,5}+1{3,6}+1{4,8}+2{1,8}+3{2,3}+4{4,7}=13.

Двудольные графы и паросочетания. Постановка задачи о назначениях

Определение. <u>Полным двудольным</u> графом называется граф с вершинами без петель, кратных и противоположных дуг (ориентация безразлична), у которого множество вершин разбито на два непересекающихся подмножества (две доли X_1 и X_2) так, что любые две вершины различных подмножеств соединены дугой, а никакие две вершины одного подмножества дугой не соединены.

Определение. Паросочетанием графа называется такой его подграф, вершины и ребра которого являются подмножествами вершин и ребер исходного графа, и степени его вершин равны 1.

Определение. Паросочетание, в котором участвуют все вершины графа, называется полным или совершенным. Очевидно, совершенное паросочетание является наибольшим и максимальным и севершенное паросочетение возможно только если в долях одинаковое количество вершин.

Задача о назначениях

Это оптимизационная задача, в которой ищется совершенное паросочетание минимального веса.

Постановка задачи. Пусть имеется *n* бригад и *n* объектов. Известна матрица стоимостей (квадратная матрица стоимостей двудольного графа) $C=(c_{ij})$, i,j=1,n, где c_{ij} – стоимость выполнения *i*-ой бригадой работ на *j*-ом объекте. Требуется так назначить бригады на объекты (по одной бригаде на один объект), чтобы суммарные затраты на работы были минимальными.

Алгоритм венгерского метода:

1. В каждой строке матрицы, не имеющей нулей, вычесть минимальный элемент из всех элементов строки (приведение матрицы по строкам).

2. В каждом столбце полученной матрицы, не имеющем нулей, вычесть минимальный из всех элементов столбца (приведение матрицы по столбцам).

3. Найти строку или столбец с наименьшим числом нулей, выделить любой из этих нулей обводкой; зачеркнуть остальные нули, стоящие в одном столбце и в одной строке с выделенным нулем; продолжить это действие, пока возможно.

4. Если после шага **3** остаются невыделенные нули, то выделяем любой из них, а в строке и столбце с выделенным нулем остальные нули зачеркиваем.

5. Если в каждой строке и в каждом столбце есть выделенный нуль, то перейти к **6,** в противном случае начинается *итерационный процесс:*

- **a.** в приведенной по строкам и столбцам матрице снимаем все выделения и зачеркивания нулей, проводим минимальное число горизонтальных и вертикальных прямых через все нули;
- **b.** среди не зачеркнутых этими прямыми чисел находим минимум;
- **c.** этот минимум вычитаем из всех не зачеркнутых чисел и прибавляем ко всем числам, которые перечеркнуты сразу двумя прямыми;
- **d.** убираем проведенные прямые и к полученной матрице применяем алгоритм, начиная с шага **3.**

6. Решение найдено, указать назначения бригад на объекты, рассчитать суммарную стоимость (затраты).

Пример. Используя венгерский метод, найти решение задачи о назначениях

Второй способ

Венгерский метод

	o1	o2	o3	o4	o5
б1	13	12	11	10	12
б2	12	13	12	12	13
б3	13	12	11	10	12
б4	11	11	10	11	14
б5	13	12	11	10	11

вычитаем миним. в каждой строке

	o1	o2	o3	o4	o5
б1	3	2	1	0	2
б2	0	1	0	0	1
б3	3	2	1	0	2
б4	1	1	0	1	4
б5	3	2	1	0	1

вычитаем миним. в каждом столбце

	o1	o2	o3	o4	o5
б1	3	1	1	$\boxed{0}$	1
б2	0	0	0	$\cancel{0}$	$\cancel{0}$
б3	3	1	1	$\cancel{0}$	1
б4	1	0	0	1	3
б5	3	1	1	$\cancel{0}$	$\boxed{0}$

выбираем ед. ноль в каждой строке,
вычеркиваем нули в столбце

	o1	o2	o3	o4	o5
б1	3	1	1	$\boxed{0}$	1
б2	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$	$\cancel{0}$	$\cancel{0}$
б3	3	1	1	$\cancel{0}$	1
б4	1	$\cancel{0}$	$\boxed{0}$	1	3
б5	3	1	1	$\cancel{0}$	$\boxed{0}$

выбираем ед. ноль в каждом столбце,
вычеркиваем нули в строке

так как не в каждой строке и столбце оказался выделенный ноль
проводим минимальное количество прямых через все нули

	o1	o2	o3	o4	o5
б1	3	1	1	0	1
~~б2~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~
б3	3	1	1	0	1
~~б4~~	~~1~~	~~0~~	~~0~~	~~1~~	~~3~~
б5	3	1	1	0	0

находим минимум среди не зачеркнутых чисел
вычитаем его из не зачеркнутых чисел и прибавляем к числам
которые перечеркнуты сразу двумя прямыми

	o1	o2	o3	o4	o5
б1	2	$\cancel{0}$	$\boxed{0}$	$\cancel{0}$	1
б2	$\boxed{0}$	$\cancel{0}$	$\cancel{0}$	1	1
б3	2	$\cancel{0}$	$\cancel{0}$	$\boxed{0}$	1
б4	1	$\boxed{0}$	$\cancel{0}$	2	4
б5	2	$\cancel{0}$	$\cancel{0}$	$\cancel{0}$	$\boxed{0}$

снова начинаем процесс выбора и вычеркивания нулей
так как в каждой строке и столбце оказался выделенный ноль,
решение найдено

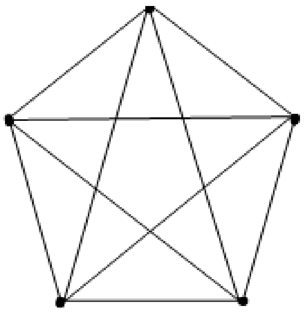
Назначения и суммарные затраты

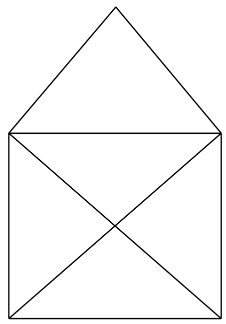
бриг	объек	затр
1	3	11
2	1	12
3	4	10
4	2	11
5	5	11
	сумма	55

Эйлеров цикл, эйлерова цепь. Условия эйлеровости и квазийлеровости графа

Определение. Цепь (цикл) называется эйлеровой (эйлеровым), если она проходит по одному разу через каждое ребро графа. Связный граф, в котором есть эйлеров цикл, называется эйлеровым. Связный граф называется квазиэйлеровым, если в нем есть эйлерова цепь.

Критерии эйлеровости и квазиэйлеровости неориентированного графа. Для того, чтобы на связном графе существовал эйлеров цикл необходимо и достаточно, чтобы степени его вершин были четными. Пример а).





а) б)

Для того, чтобы на связном графе существовала эйлерова цепь необходимо и достаточно, чтобы он имел ровно две вершины нечетной степени, причем эйлерова цепь всегда соединяет вершины нечетной степени. Пример б).

Суть алгоритмов отыскания эйлерова цикла: разбить граф на простые циклы с непересекающимися множествами ребер и пересекающимися множествами вершин. Склеивание (объединение) этих циклов через общие вершины дает эйлеров цикл.

Чтобы найти Эйлерову цепь, нужно соединить две вершины с нечетными степенями фиктивным ребром. Тогда задача сводится к нахождению Эйлерова Из найденного цикла удаляется фиктивное ребро, тем самым находится искомая Эйлерова цепь.

Пример (вариант геометрического решения). Найти эйлерову цепь в неориентированном графе G, изображенном на рис. 6.

Решение. В рассматриваемом графе нечетные степени имеют вершины v_3 и v_1 (степень этих вершин равна 3). Соединяя эти вершины фиктивным ребром так, как показано на рисунке, получаем граф G', рис.7.

Поскольку в конечном итоге будет получена цепь, то очевидно, что началом и концом этой цепи будут вершины с нечетными степенями. Следуя описанному выше алгоритму, будем искать циклы μ_i так, чтобы хотя бы один из них начинался или кончался на вершинах v_3 или v_1.

Пусть цикл μ_1 составят ребра, проходящие через следующие вершины: $v_3\ v_4\ v_7\ v_6\ v_1\ v_2\ v_3$ (окрашен оранжевым цветом). Согласно алгоритму, удаляем из G' все ребра, задействованные в цикле μ_1.

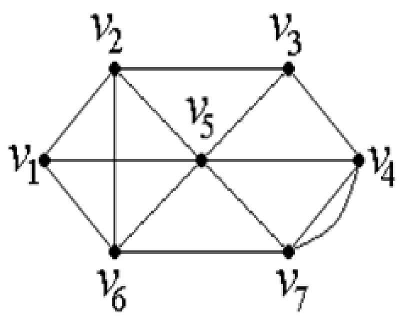


Рисунок 5

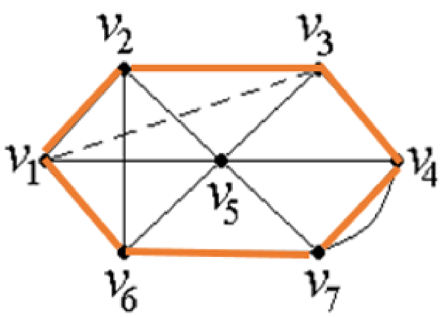


Рисунок 6

Теперь граф G' будет таким, как показано на следующем рис.8.

Составляем следующий цикл μ_2: $v_4\ v_5\ v_6\ v_2\ v_5\ v_7\ v_4$. Граф G' после удаления ребер, составляющих цикл μ_2, изображен на рис.9.

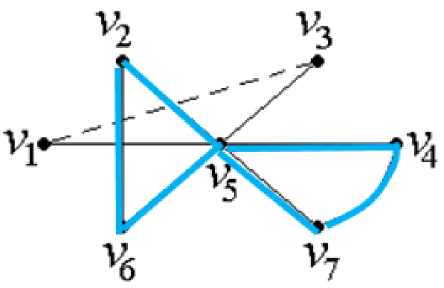


Рисунок 7

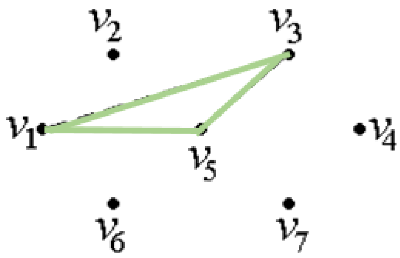


Рисунок 8

Очевидно, что последний цикл μ_3 (рис.9) будет состоять из v_3 v_5 $v_1|v_3$, где последнее ребро, соединяющее вершины v_1 и v_3 – фиктивно. После удаления ребер, составляющих цикл μ_3, в графе G' не останется ни одного ребра.

Теперь по общим вершинам склеиваем полученные циклы. Поскольку μ_1 и μ_2 имеют общую вершину v_4, то, объединяя их, получим следующий цикл: $v_3\ v_4\ v_5\ v_6\ v_2\ v_5\ v_7\ v_4\ v_7\ v_6\ v_1\ v_2\ v_3$. Теперь склеим получившийся цикл с циклом μ_3: $v_3\ v_4\ v_5\ v_6\ v_2\ v_5\ v_7\ v_4\ v_7\ v_6\ v_1\ v_2\ v_3\ v_5\ v_1|v_3$. Удаляя фиктивное ребро, получаем искомую Эйлерову цепь: $v_3\ v_4\ v_5\ v_6\ v_2\ v_5\ v_7\ v_4\ v_7\ v_6\ v_1\ v_2\ v_3\ v_5\ v_1$.

Гамильтоновы графы

Определение. Цепь (цикл) называется гамильтоновой (гамильтоновым), если она (он) проходит через каждую вершину графа ровно один раз. Связный граф называется гамильтоновым, если он содержит гамильтонов цикл.

Термин «гамильтонов» происходит от имени известного ирландского математика 19 в. У. Гамильтона. Практическое применение гамильтоновых циклов связано со знаменитой задачей коммивояжера.

Постановка задачи коммивояжёра

Задача состоит в следующем: коммивояжер (бродячий торговец) должен посетить каждый из заданных городов по одному разу, выехав из некоторого города и вернувшись в него же, при этом стоимость (или длина пути) поездки должна быть минимальной.

Решение задачи коммивояжера сводится к отысканию на взвешенном графе гамильтонова цикла минимального веса.

Задача коммивояжера – важная задача транспортной логистики, отрасли, занимающейся планированием транспортных перевозок.

Универсальные языки V^* и V^+. Лексикографический номер слова универсального языка по заданному алфавиту

Будем обозначать V^* - множество всех слов в данном алфавите, включая пустую цепочку, обозначаемую $ или λ. Длину слова $v \in V^*$ обозначим через $|v|$, тогда $|\$|=0$, $|abbca| = 5$.

Обозначим множество V^+ всех слов, исключая пустое слово, в данном алфавите V.

Пусть $V = \{a_1, a_2, ..., a_n\}$ - алфавит. В лексикографической нумерации пустому слову присваивается номер 0; буквам $a_1, a_2, ..., a_n$ - соответственно номера 1,2,…,n. Лексикографический номер $l(\alpha)$ слова $\alpha = a_{i_k} a_{i_{k-1}} ... a_{i_1}$ длины k вычисляется по формуле

$$l(\alpha) = n^{k-1} i_k + n^{k-2} i_{k-1} + ... + n^0 i_1 .$$

Пример. Пусть дан алфавит $V = \{a, b, c\}$. Перечислим элементы универсального языка V^+ в лексикографическом порядке $V^+ = \{a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, ...\}$. Требуется выполнить следующие задания.

1)Найти лексикографический номер $l(\alpha)$ слова $\alpha = aabca$.

2) Сделать проверку, восстановив по номеру слово.

3) Перечислить в лексикографическом порядке 10 слов, следующих за заданным словом $\alpha = aabca$.

Решение.

1) Имеем мощность алфавита n=3, длина слова k=5, номера по алфавиту значений разрядов $(i_k, i_{k-1}, \ldots i_1) = (1,1,2,3,1)$, следовательно, $l(\alpha)=3^4\cdot1+3^3\cdot1+3^2\cdot2+3\cdot3+3^0\cdot1=136$.

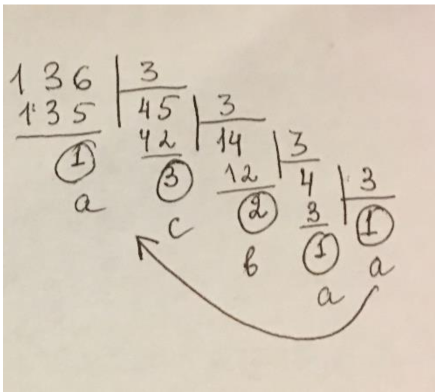


Рисунок 9

2) Номер слова 136. Делением уголком восстанавливаем слово, см. рис. 10

3) Выпишем в лексикографическом порядке слова $\alpha = aabca,\quad aabcb,\quad aabcc,\quad aacaa,\quad aacab,$
$aacac,\quad aacba,\quad aacbb,\quad aacbc,\quad aacca,\quad aaccb = \beta$.

Проверим, вычислив номер последнего слова: $l(\beta)=3^4\cdot1+3^3\cdot1+3^2\cdot3+3\cdot3+3^0\cdot2=146$.

Понятие формальной грамматики, правил и вывода в грамматике

Определение: Формальной грамматикой Г (может обозначаться G) называется совокупность четырех составляющих : Г = ⟨V, W, I, R⟩, где

V- терминальный алфавит (может обозначаться T,X); буквы этого алфавита называются терминальными символами или терминалами; из них строятся слова (цепочки), порождаемые грамматикой; для обозначения букв терминального словаря в дальнейшем условимся использовать строчные буквы латинского алфавита или цифры, а для обозначения слов – греческие буквы;

W – нетерминальный (может обозначаться N), вспомогательный алфавит, причем

$V \cap W = \varnothing$; буквы этого алфавита – нетерминалы используются при построении цепочек; они обозначают какую-либо сущность языка, например, команду; они могут входить в промежуточные цепочки, но не должны входить в результат построения; условимся для обозначения нетерминальных символов использовать прописные буквы латинского или русского алфавита;

I - начальный или стартовый символ, аксиома грамматики $I \in W$ (может обозначаться S), с которого начинаются все выводы.

R - множество правил вывода (продукций) (может обозначаться P) или порождающих правил; при символической схеме правила записываются в виде $\alpha \to \beta$, где α, β - цепочки из $(V \cup W)^*$, стрелка означает замену вхождения одной цепочки на другую, при этом левая часть правила обязательно содержит хотя бы один нетерминал.

Определение. Выводом слова $\gamma \in V^*$ в грамматике Г называется последовательность цепочек, при этом каждая последующая цепочка последовательности получается из предыдущей с помощью одного из правил вывода.

Определение. Множество всех конечных цепочек терминального алфавита V грамматики Г, выводимых из начального символа I, называется *языком,* порождаемым грамматикой Г, и обозначается L(Г).

Определение: Рекурсивным называется правило, содержащее один и тот же нетерминал в левой и правой частях правила.

Определение: Если язык, порождаемый грамматикой Г, не содержит ни одного слова конечной длины, то он называется *пустым.*

Пример. Найти язык, порождаемый грамматикой.

Г: V= {a, b}, W = {I, A}, R = {I $\to$ aA|\$, A $\to$ bA|a}.

Решение. Выпишем последовательность цепочек. Цепочки, содержащие нетерминалы, являются промежуточными. В рамочку обводим окончательные цепочки, принадлежащие языку.

$$\mathrm{I} \Rightarrow \left[\begin{array}{l}\mathrm{aA}\\ \boxed{\$}\end{array}\right. \Rightarrow \left[\begin{array}{l}\mathrm{abA}\\ \boxed{aa}\end{array}\right. \Rightarrow \left[\begin{array}{l}\mathrm{abbA}\\ \boxed{aba}\end{array}\right. \Rightarrow \ldots \left[\begin{array}{l}\mathrm{ab}^{n+1}\mathrm{A}\\ \boxed{ab^n a}\end{array}\right.,\ldots..$$

Получим язык L={\$}$\cup\{ab^n a, n = 0,1,2 \ldots..\ \}$. Это регулярный язык, так как правила грамматики имеют вид, соответствующий регулярной грамматике.

Понятие конечного автомата-распознавателя

КА-анализатор (распознаватель) можно *интерпретировать как устройство*, работающее по тактам, состоящее из полубесконечной ленты, в ячейках которой находятся символы слова во входном алфавите X; считывающего устройства (СУ), обозревающего на каждом такте(шаге) одну ячейку ленты; устройства управления (УУ), переключение состояний которого зависит от заданной функции переходов. Состояние УУ на следующем шаге определяются состоянием и обозреваемым символом на предыдущем шаге. Количество состояний УУ конечно и определяется множеством Q.

Формальное определение КАА включает 5 составляющих

$$KAA = \langle X, Q, q_н, F, \delta \rangle,$$

где X - входной алфавит, совпадающий с терминальным алфавитом V грамматики;

$Q = \{q_1, q_2, ..., q_n\}$ - алфавит состояний;

$I \subset Q$ или $q_н \subset Q$ - выделенное начальное состояние;

$F \subset Q$ - выделенное множество заключительных состояний;

δ – функция, переходов, сопоставляет каждой паре (состояние, буква)=(q_i,a_j) некоторое состояние q_k.

Определение. Слово языка $\alpha = a_{i_1} a_{i_2} \ldots a_{i_k}$ *распознается автоматом,* если, начиная работу из начального состояния $q_н$ и обозревая первый символ слова a_{i1}, за конечное число шагов автомат переходит к последнему символу слова a_{ik}, при этом оказывается в одном из заключительных состояний F.

Определение. Автомат распознает язык, если он распознает все слова языка и только их.

Функцию переходов можно представить набором команд, задать с помощью таблицы или с помощью графа переходов.

В таблице переходов обычно строки соответствуют состояниям, столбцы – символам входного алфавита (может быть наоборот). На пересечении i –й стороки и j – го столбца записываются состояние q_k, что означает: если УУ на данном такте находится в состоянии q_i и СУ обозревает символ «a_j», то УУ переходит на следующем такте в состояние q_k и СУ сдвигается на следующую справа ячейку.

Соответствующую команду можно записать в виде (q_i,a_j)→q_k.

В графе переходов узлы (кружки) - состояния КА, дуги показывают переходы между состояниями, каждая дуга размечена одним символом входного алфавита. Запись букв через «,» или через «+» (операция «или») означает кратные дуги графа.

Пример. Функция переходов КА задана таблицей. Изобразить граф переходов, если $F = \{q4\}$, $q_н = q1$. Записать несколько первых слов языка в лексикографическом порядке, распознаваемые этим автоматом.

Q\X	*a*	*b*
q_1	q_3, q_1	q_3
q_2	q_4	$q2$
q_3	-	$q4$
$\boxed{q_4}$	-	q_2

В таблице заключительные состояния условимся обводить в рамку.

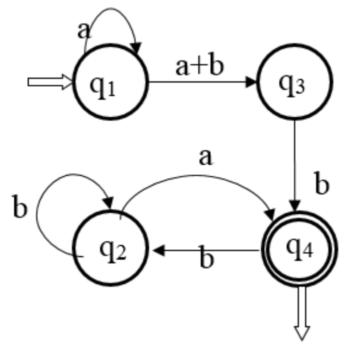


Рисунок 10

На графе (рис.11) начальное и заключительные состояния обведены в двойной кружочек, и имеют утолщенные стрелки, показывающие, какое состояние начальное, а какие заключительные. Петли на графе соответствуют рекурсивным правилам. Сколь угодно длинные цепочки образуются за счет рекурсивных правил и за счет замкнутых путей (в данном случае между состояниями q2, q4).

По графу переходов определяем слова, распознаваемые автоматом: $\{ab, bb, aab,.abb, aaab.....\}$

Синтез КА по регулярной грамматике

Каждой регулярной грамматике можно поставить в соответствие КА, который распознает язык, порожденный этой грамматикой, и только его. Будем рассматривать только праволинейную грамматику, в которой допустимы правила A→a, A→aB.

Пусть грамматика и соответствующий ей автомат обозначены:

$$Г = \langle V, W, I, R\rangle \qquad и \qquad КА = \langle X, Q, q_н, F, \delta\rangle.$$

Установим связь между алфавитами грамматики и автомата:

– терминальный алфавит грамматики V является входным алфавитом X конечного автомата;
– начальное состояние автомата $q_н$ соответствует аксиоме грамматики I;
– множество состояний Q автомата соответствует нетерминальному алфавиту W, пополненному множеством заключительных состояний $Q = W \cup F$. Количество состояний |Q| не меньше числа нетерминалов |W| и определяется структурой языка;
– система команд функции переходов автомата формируется следующим образом:

правило грамматики A→aB переходит в команду Aa→B таблицы переходов, где A,B∈Q, a∈X;

правило грамматики A→a переходит в команду Aa→q_F, где q_F- заключительное состояние;

правило грамматики A→$ означает, что A –заключительное состояние, т.е. A= q_F.

Замечание. Для обозначений состояний автомата можно использовать символы нетерминального алфавита, буквы q с индексами либо просто цифры.

Пример. Найти автомат, распознающий язык, порождаемый грамматикой.

Г: V= {a, b}, W = {I, A}, R = {I → aA|$, A → bA|a}.

Решение. Выпишем команды функции переходов, соответствующие правилам грамматики, используя описанным выше закономерностям:

δ: $Ia \to A$, $I \to q_{F1}$, $Ab \to A$, $Aa \to q_{F1}$.

Заключительные состояния: $I = q_{F1}$, q_{F2}.

Изобразим граф переходов и составим таблицу переходов (рис. 12)

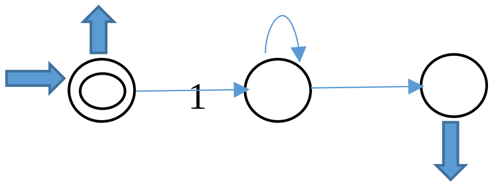


Рисунок 11

Q\X	2	1	3
I	-	A,I	B
A	A	B	-
B	-	q_F	B
q_{F2}	-	-	-

.

Понятие автомата с выходами (преобразователя)

КА с выходами можно интерпретировать *как устройство*, работающее по тактам, состоящее из полубесконечных входной и выходной лент, разбитых на ячейки, считывающего и записывающего устройства (СЗУ), устройства управления (УУ). На каждом такте УУ находится в одном из конечного множества состояний, а СЗУ считывает символ из ячейки входной ленты. В зависимости от состояния и выходного символа СЗУ записывает символ в ячейку выходной ленты, а УУ переходит в новое состояние.

Конечный автомат с выходами можно интерпретировать *как преобразователь* (КАП) или или отображение (оператор) $y = S(x)$, преобразующий каждое входное слово $x = x_{i_1} x_{i_2} \ldots x_{i_n}$ в выходное $y = y_{i_1} y_{i_2} \ldots y_{i_n}$ той же длины. Входное слово – последовательность букв, поступающих на вход автомата в дискретные моменты времени (такты) $t = 1,2,3\ldots$Выходное слово – последовательность букв, выдаваемых автоматом.

Формальное определение КАП включает 6 составляющих

$КАП = \langle X, Y, Q, q_н, \delta, \lambda \rangle$, где

X, Y - входной и выходной алфавиты;

$Q = \{q_1, q_2, \ldots, q_n\}$ - алфавит состояний;

$\delta: \quad Q \times X \to Q$ – функция переходов;

$\lambda: \quad Q \times X \to Y$ – функция выходов;

$q_н \in Q$ - выделенное начальное состояние.

Замечание. В конечных автоматах с выходами обычно не выделяется множество заключительных состояний. В качестве заключительного может быть любое состояние из множества Q.

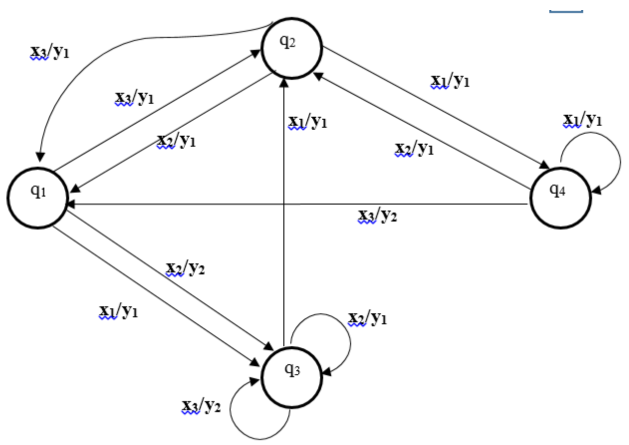
Способы задания автоматов. Существуют различные способы задания автоматов: табличный , графический и функциональный.

Таблица переходов — табличное представление функций переходов и выходов. Обычно в такой таблице каждой строке соответствует одно состояние, а столбцу — один допустимый входной символ. В ячейке на пересечении строки и столбца записывается пара «состояние, выходной символ». В таблице переходов могут строкам соответствовать символы алфавита, а столбцам – состояния.

В графе переходов вершинам графа соответствуют состояния, дуги помечаются парой «вх.символ|вых символ».
Заметим, что КАП - детерминированное устройство, т.е из данного состояния возможен переход только в одно из состояний.

Пример. КАП задан алфавитами и таблицей переходов $X=\{x_1,x_2,x_3\}$, $Y=\{y_1,y_2\}$, $Q=\{q_1,q_2,q_3,q_4\}$.
Нарисовать граф переходов. По входному слову $x_1x_1x_2x_3$ найти выходное слово, если начальное состояние автомата - q_1, а в качестве заключительных возможны все состояния.

Решение. На рис. 12 изображен соответствующий граф переходов.



Q\X	x_1	x_2	x_3
q_1	q_3, y_1	q_3, y_2	q_2, y_1
q_2	q_4, y_1	q_1, y_1	q_1, y_1
q_3	q_2, y_1	q_3, y_1	q_3, y_2
q_4	q_4, y_1	q_2, y_1	q_1, y_2

Рисунок 12

При выводе символов выходного слова по тактам рассматриваем на каждом такте пару (состояние, вход) и находим по таблице, какому выходному символу это соответствует:

$$S(q_1,x_1x_1x_2x_3)=y_1S(q_3,x_1x_2x_3)=y_1y_1S(q_2,x_2x_3)=y_1y_1y_1S(q_1,x_3)=y_1y_1y_1y_1$$

Автоматы Мили и Мура

В зависимости от способа формирования выходного сигнала различают автоматы первого и второго рада или автоматы Мили и Мура. В автомате Мили выходной символ зависит от входного символа и от состояния, в котором находится автомат. Автомат Мура – частный случай автомата Мили, у которого выходной символ не зависит от входного, а только от состояния, т.е. функция выходов зависит от только от состояний. В таблице переходов (графе переходов) автомата Мура символы состояний и соответствующие выходные символы записываются в одной клетке (в одной вершине) или в смежных клетках.

X\Y	Y_1	Y_2	Y_2
Q	q_1	q_2	q_3
X_1	1	3	2
X_2	2	3	2

Пример. Дана таблица переходов автомата Мура S. Составить граф переходов и найти $S(q_2,x_1x_2x_1)$. Считать все состояния заключительными.

Решение. На рис. 13 изображен соответствующий граф переходов.

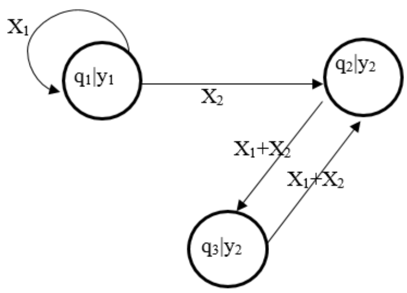


Рисунок 13

$$S(q_2,x_1x_2x_1)=y_2S(q_3,x_2x_1)=y_2y_2S(q_2,x_1)=y_2y_2y_2S(q_3)=y_2y_2y_2$$

ПЕРЕЧЕНЬ ТЕОРЕТИЧЕСКИХ ВОПРОСОВ ДЛЯ ПОДГОТОВКИ К ЭКЗАМЕНУ

1. Понятие графа и орграфа. Инцидентность и смежность вершин и ребер. Степени вершин. Теорема Эйлера о степенях вершин.
2. Виды графов. Планарность.
3. Матрицы смежности и инцидентности для орграфа и неориентированного графа, их свойства.
4. Пути, контуры, маршруты, циклы.
5. Нахождение числа путей из i–й в j–ю вершину с заданным числом дуг через матрицу смежности.
6. Связность неориентированного графа. Сильная, односторонняя и слабая связность орграфа.
7. Взвешенный граф, его матрица. Смысл меток $\lambda_i^{(k)}$. Алгоритм Дейкстры.
8. Графы–деревья. Свойства деревьев. Понятие остова.
9. Алгоритм Краскала отыскания остова минимального веса на взвешенном графе.
10. Двудольные графы, их представление матрицами. Наибольшее и совершенное паросочетания.
11. Постановка задачи о назначениях, ее трактовка в терминах графов. Венгерский метод.
12. Эйлеров и квазиэйлеров графы. Условия эйлеровости графа. Алгоритм отыскания эйлерова цикла и эйлеровой цепи.
13. Полный граф. Гамильтонов цикл. Постановка задачи коммивояжера.
14. Универсальные языки V^* и V^+. Лексикографический номер слова универсального языка по заданному алфавиту.
15. Восстановление слова по лексикографическому номеру в универсальном языке.
16. Понятие формальной грамматики, ее составляющие. Вывод в формальной грамматике.
17. Понятие формального языка. Рекурсивные правила. Пустой язык.
18. Понятие конечного автомата-распознавателя.
19. Таблица и граф переходов конечного автомата.
20. Условия распознавания слова автоматом
21. Синтез конечного автомата по регулярной грамматике. Соответствие между правилами грамматики и командами конечного автомата.
22. Понятие автомата с выходами (преобразователя). Формальное описание преобразователя.
23. Таблица и граф переходов преобразователя.
24. Автоматы Мили и Мура.

СПИСОК ИНФОРМАЦИОННЫХ РЕСУРСОВ

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Кузнецов, О.П.Дискретная математика для инженера [Электронный ресурс] / О.П. Кузнецов ; Кузнецов О. П. - 6-е изд., стер. - Санкт-Петербург : Лань, 2022. - 400 с.
2. Алымова, Е.В.Конечные автоматы и формальные языки [Электронный ресурс] : учебник / Е.В. Алымова, В.М. Деундяк, А.М. Пеленицын ; Е. В. Алымова, В. М. Деундяк, А. М. Пеленицын. - Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2018. - 292 с.

ИНТЕРНЕТ-ИСТОЧНИКИ

1. Образовательный портал ДГТУ : [сайт]. URL: http://skif.donstu/
2. Информационная система "Единое окно доступа к образовательным ресурсам" : [сайт]. URL: http://window.edu.ru/
3. Образовательный математический сайт по математике и программированию : [сайт]. URL: http://old.exponenta.ru/