РЕФЕРАТ

Целью данной курсовой работы является исследование производящих функций, применимых к графам. Вопросы, требующие рассмотрения по данной теме, были полностью разобраны в данной курсовой работе. Пояснительная записка, общем объёмом 21 стр., содержит 2 рисунка, 4 использованных источника.

Ключевые слова: производящие функции, теория графов, комбинаторика, аналитические методы, структурные свойства, перечисление, решение задач.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В современной теории графов, производящие функции являются важным инструментом для анализа и перечисления комбинаторных структур. Они позволяют формализовать и изучать различные объекты, включая графы, с помощью математических методов. В данной статье мы рассмотрим основные принципы использования производящих функций в теории графов, исследуем их применение для решения разнообразных задач и проведем анализ полученных результатов.

1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ГРАФОВ

В теории графов существует множество основных понятий, которые используются для анализа и описания свойств графов. Некоторые из основных понятий теории графов включают:

- Граф – абстрактная математическая структура, состоящая из вершин (узлов) и рёбер (дуг), соединяющих вершины;
- Ориентированный граф – граф, в котором каждое ребро имеет направление, указывающее на направление связи между вершинами;
- Неориентированный граф – граф, в котором рёбра не имеют направления, то есть связь между вершинами двусторонняя;
- Подграф – часть графа, образованная некоторыми его вершинами и рёбрами;
- Связный граф – граф, в котором любые две вершины можно соединить путём последовательного прохода по рёбрам (также называется полным графом);
- Дерево – связный ациклический граф, то есть граф без циклов;
- Путь – последовательность вершин и рёбер, соединяющая вершины в графе;
- Цикл – путь, который начинается и заканчивается в одной и той же вершине;
- Степень вершины – количество рёбер, инцидентных вершине, то есть входящих или выходящих из данной вершины;
- Матрица смежности - матрица, используемая для представления связей между вершинами в графе;
- Компонент связанности – это максимальный подграф, в котором любые две вершины соединены путем. Граф может содержать несколько компонент связанности, если он состоит из нескольких независимых подграфов, между которыми нет ребер;

Пример ориентированного графа представлен на рис. 1, матрица смежности для этого графа – на табл. 1.

На рис. 1, вершины 3, 4, 6, 7 образуют подграф, вершины 1-8 образуют одну компоненту связанности, а 9-11 – другую; подграф, состоящий из этих вершин, является деревом.

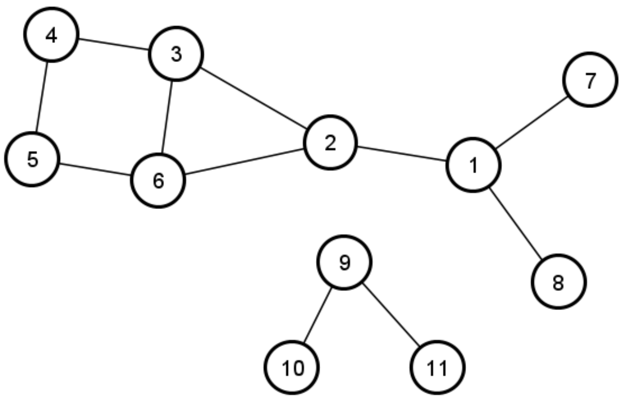

Рис. 1

	1	2	3	4	5	6	7	8	9	10	11
1	0	1	0	0	0	0	1	1	0	0	0
2	1	0	1	0	0	1	0	0	0	0	0
3	0	1	0	1	0	1	0	0	0	0	0
4	0	0	1	0	1	0	0	0	0	0	0
5	0	0	0	1	0	1	0	0	0	0	0
6	0	1	1	0	1	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	1
10	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	0	0	0	0	0	1	0	0

Табл. 1

Производящая функция — это мощный инструмент, который используется для изучения комбинаторных структур, таких как графы. Производящая функция представляет собой формальный степенной ряд, в котором коэффициенты перед степенями переменной соответствуют количеству комбинаторных объектов определённого размера или свойства.

Вот некоторые основные свойства производящих функций:

1. Линейность: если $G(x)$ и $H(x)$ - производящие функции для двух графов G и H соответственно, то производящая функция для объединения (дизъюнктивного объединения) графов G и H будет равна $G(x) + H(x)$;
2. Конволюция: если $G(x)$ и $H(x)$ - производящие функции для двух графов G и H соответственно, то производящая функция для свертки (произведения) графов G и H будет равна $G(x) \cdot H(x)$;

3. Инверсия: если $G(x)$ - производящая функция для графа ы, то производящая функция для дополнения графа G будет равна $\frac{1}{1-G(x)}$;
4. Производные: различные производные производящей функции могут использоваться для вычисления различных комбинаторных параметров графа, таких как количество вершин, ребер, компонентов связанности и т.д.;
5. Сложение вершин: производящая функция может быть использована для анализа графов с различными свойствами вершин, такими как степени вершин, раскраски вершин и другими характеристиками;
6. Сложение ребер: производящая функция также позволяет анализировать графы с различными свойствами ребер, такими как веса ребер, циклы, пути и другие комбинаторные параметры.

Эти свойства производящих функций в теории графов позволяют эффективно моделировать и анализировать различные комбинаторные структуры и задачи, связанные с графами.

Теория графов и производящие функции тесно связаны друг с другом. Вот несколько способов, которыми теория графов и производящие функции взаимодействуют:

1. Перечисление структур: производящие функции часто используются для перечисления различных комбинаторных структур, таких как деревья, пути, циклы и, конечно же, графы. С их помощью можно эффективно подсчитывать количество различных графов определенного размера, с определенными свойствами и т.д.;
2. Анализ свойств графов: производящие функции позволяют анализировать различные свойства графов, такие как степенные последовательности, циклические структуры, соединенные

компоненты и т.д. Это помогает понять структурные особенности графов и обобщить результаты на более общие классы графов;

3. Рекурсивные соотношения: многие комбинаторные структуры, включая графы, могут быть описаны с помощью рекурсивных соотношений. Производящие функции позволяют элегантно формализовать эти соотношения и решать их с помощью методов аналитической теории;
4. Комбинаторный анализ: производящие функции являются мощным инструментом для комбинаторного анализа различных структур и процессов. Они позволяют решать задачи на подсчет комбинаторных объектов, нахождение вероятностей различных событий и т.д..

Таким образом, производящие функции играют важную роль в теории графов, предоставляя математический аппарат для анализа и решения различных задач, связанных с графами и другими комбинаторными структурами.

2. ПРОИЗВОДЯЩИЕ ФУНКЦИИ ДЛЯ РАЗЛИЧНЫХ ТИПОВ ГРАФОВ

Для разных типов графов используются разные версии производящих функций, потому что структура графа и его свойства могут существенно отличаться в зависимости от типа графа.

Более конкретно, производящая функция в простых графах может быть определена следующим образом: пусть $G(x)$ — это производящая функция для последовательности чисел g_n, где g_n – количество различных простых графов на n вершинах. Тогда производящая функция $G(x)$ будет иметь вид:

$$G(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3 + \dots$$

Производящая функция для ориентированных графов представляет собой формальный степенной ряд, который используется для описания комбинаторных свойств ориентированных графов. В отличие от производящей функции для простых графов, производящая функция для ориентированных графов учитывает направленность рёбер и может использоваться для подсчёта различных комбинаторных характеристик таких графов. Производящая функция в ориентированных графах может быть определена следующим образом: пусть $D(x)$ — это производящая функция для последовательности чисел d_n, где d_n — количество различных ориентированных графов на n вершинах. Тогда производящая функция $D(x)$ будет иметь вид:

$$D(x) = d_0 + d_1 x + d_2 x^2 + d_3 x^3 + \dots ,$$

где коэффициенты d_n соответствуют количеству различных ориентированных графов на n вершинах. С помощью этой производящей функции можно решать различные задачи, связанные с перечислением и анализом ориентированных графов, такие как подсчёт числа различных графов определённого размера или типа, исследование свойств ориентированных графов.

Для взвешенного графа, в котором каждому ребру присвоено некоторое числовое значение (вес), можно определить производящую функцию, которая будет учитывать веса ребер. Для этого можно использовать понятие производящей функции для графов с взвешенными ребрами.

Предположим, что у нас есть взвешенный граф с вершинами $v_1, v_2, \dots, v_n$ ребрами с весами v_{ij} между вершинами v_i и v_j. Тогда производящая функция для такого графа может быть определена как формальный степенной ряд:

$$G(x) \ = \ \sum_{n=0}^{\infty} \sum (i_1, i_2, \dots, i_n, w_{i_1 i_2}, w_{i_2 i_3}, \dots, w_{i_{n-1} i_n}) \, x^n$$

где w_{ij} — вес ребра между вершинами v_i и v_j, а суммирование происходит по всем возможным последовательностям вершин $i_1, i_2, \dots i_n$.

Таким образом, производящая функция для взвешенного графа учитывает все возможные пути и циклы в графе с учетом весов ребер. Это позволяет анализировать различные свойства взвешенных графов, такие как суммарные веса путей, минимальные или максимальные пути и т.д.

Производящие функции для взвешенных графов могут быть использованы для решения различных задач, связанных с оптимизацией путей, поиска кратчайших путей, анализа структуры графа и других комбинаторных задач, которые учитывают веса ребер.

Производящая функция для графа-дерева может быть построена следующим образом. Пусть $T(z)$ - производящая функция для деревьев, где коэффициент при z^n равен количеству вершин в дереве с n вершинами. Тогда для графа-дерева производящая функция будет выглядеть следующим образом:

$$D(z) = \sum_{n=1}^{\infty} T(z)^{\frac{n}{n!}}$$

Здесь $T(z)$ - производящая функция для деревьев, а $D(z)$ - производящая функция для графа-дерева.

Производящая функция для деревьев может быть выражена через саму себя следующим образом:

$$T(z) = z + z * T(z)^2$$

Это уравнение отражает тот факт, что дерево может быть представлено как одна вершина или как несколько деревьев, соединенных в корневую вершину. Решив это уравнение, можно получить явное выражение для производящей функции деревьев $T(z)$, которое затем можно подставить в формулу для производящей функции графа-дерева $D(z)$.

Производящая функция для графа с циклами может быть построена следующим образом. Пусть $C(z)$ - производящая функция для графов с циклами, где коэффициент при z^n равен количеству вершин в графе с циклами, содержащем n вершин.

Для графов с циклами можно использовать следующее рекуррентное соотношение:

$$C(z) = z + \frac{1}{2} z^2 C(z)^2$$

Это уравнение отражает тот факт, что граф с циклами может быть представлен как одна вершина или как два графа с циклами, соединенные ребром.

Решив это уравнение, можно получить явное выражение для производящей функции графов с циклами $C(z)$. Это позволит нам легко вычислять различные комбинаторные характеристики таких графов, используя производящую функцию.

Давайте рассмотрим пример графа (рис. 2), представленного матрицей смежности:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

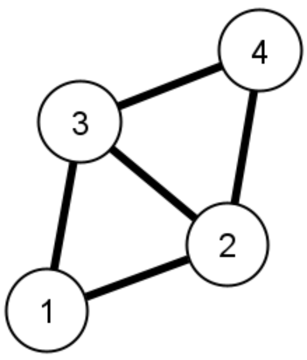

Рис. 2

Эта матрица описывает граф с четырьмя вершинами и ребрами, соединяющими вершины. Теперь мы можем использовать эту матрицу для построения производящей функции для данного графа.

Для данного графа мы можем выразить производящую функцию через матрицу смежности следующим образом:

$$C(z) = z + \frac{1}{2}z^2(C(z) * A^2)$$

Где A^2 — это квадрат матрицы смежности A, который представляет количество путей длины 2 между каждой парой вершин в графе.

Подставляя A^2 в уравнение, мы можем найти производящую функцию для данного графа.

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

Теперь мы можем подставить A^2 в уравнение для производящей функции:

$$C(z) = z + \frac{1}{2}z^2 * [C_1(z), C_2(z), C_3(z), C_4(z)] * \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 3 & 2 & 1 \\ 2 & 2 & 3 & 2 \\ 1 & 1 & 2 & 2 \end{bmatrix}$$

Где $C_i(z)$ — это производящая функция для вершины i.

Мы можем далее упростить это уравнение и найти производящую функцию для данного графа.

Таким образом, производящая функция для данного графа равна:

$$C(z) = z + z^2\big(C_1(z) + C_3(z)\big) + \frac{1}{2}z^2(C_2(z) + C_4(z))$$

Производящая функция для вершины в случае графов обычно определяется следующим образом: пусть у нас есть вершина, не имеющая инцидентных рёбер. Тогда производящая функция для такой вершины будет представлять собой просто переменную z, так как вершина не имеет соседей.

Если же вершина инцидентна рёбрам, то мы можем выразить производящую функцию для такой вершины через производящие функции смежных с ней вершин. Например, если у вершины есть k инцидентных рёбер, то производящая функция для такой вершины будет содержать слагаемое z^k, умноженное на произведение производящих функций смежных вершин. Таким образом, производящая функция для вершины зависит от её степени и от свойств графа, в котором она находится.

3. ПРИМЕНЕНИЕ ПРОИЗВОДЯЩИХ ФУНКЦИЙ В ТЕОРИИ ГРАФОВ

Производящие функции в теории графов могут использоваться для решения различных задач, связанных с графами. Некоторые из таких задач включают:

1. Подсчет количества различных графов заданного размера или с определенными свойствами. Производящие функции могут помочь найти аналитическое выражение для числа графов определенного типа.
2. Вычисление средних характеристик графов, таких как средняя степень вершин, средняя длина пути между вершинами и другие метрики.
3. Решение задач комбинаторной оптимизации на графах, таких как поиск наименьшего остовного дерева.
4. Исследование свойств случайных графов и вероятностных моделей графов. Производящие функции могут помочь оценить вероятность появления определенных структур в случайных графах.
5. Анализ процессов на графах, таких как случайные блуждания, распространение информации и другие динамические процессы.

Рассмотрим задачу на подсчет количества всех возможных простых графов на n вершинах. Простой граф — это граф, в котором нет петель (ребер, соединяющих вершину с самой собой) и кратных ребер (несколько ребер между одной и той же парой вершин).

Для решения этой задачи мы можем использовать производящую функцию для простых графов. Пусть a_n — количество всех простых графов на n вершинах. Тогда производящая функция $A(x)$ для последовательности a_n будет иметь вид: $A(x) = \sum_{n=0}^{\infty} a_n x^n$.

Теперь, чтобы найти аналитическое выражение для производящей функции $A(x)$, рассмотрим следующие шаги:

Каждый простой граф на n вершинах можно представить как объединение двух множеств: множества ребер и множества подмножеств вершин, которые соединены этими ребрами.

Множество ребер можно описать производящей функцией для всех возможных комбинаций ребер. Пусть $B(x)$ - производящая функция для всех возможных комбинаций ребер.

Множество подмножеств вершин можно описать производящей функцией для всех возможных комбинаций подмножеств вершин. Пусть $C(x)$ - производящая функция для всех возможных комбинаций подмножеств вершин.

Произведение производящих функций $B(x)$ и $C(x)$ даст нам производящую функцию для всех простых графов на n вершинах.

Таким образом, использование производящих функций позволяет нам эффективно подсчитать количество различных простых графов заданного размера.

Чтобы подсчитать количество различных возможных простых графов на n вершинах с помощью полученной производящей функции, мы можем воспользоваться формулой для коэффициента при x^n в разложении произведения производящих функций.

Пусть у нас есть производящая функция $A(x) = B(x) \cdot C(x)$, где $B(x)$ и $C(x)$ - производящие функции для множеств ребер и подмножеств вершин соответственно. Тогда коэффициент при x^n в разложении произведения $A(x)$ будет представлять собой количество различных простых графов на n вершинах.

Для вычисления коэффициента при x^n в разложении произведения производящих функций можно воспользоваться формулой коэффициента при произведении степенных рядов. Для этого нужно выразить произведение производящих функций как одну производящую функцию и затем найти коэффициент при x^n.

Итак, после нахождения коэффициента при x^n в разложении произведения производящих функций, мы получим количество различных возможных простых графов на n вершинах.

Анализ алгоритмов на графах с использованием производящих функций зачастую связан с исследованием комбинаторных структур, связанных с графами, и анализом их свойств. Вот несколько способов, которыми производящие функции могут быть полезны при анализе алгоритмов на графах:

1. Количество путей и циклов: производящая функция может использоваться для определения количества путей и циклов определенной длины в графе, что может быть полезным при анализе алгоритмов поиска путей или циклов. Например, можно использовать производящую функцию для подсчета количества всех путей или циклов определенной длины, и затем использовать эту информацию для оценки эффективности алгоритма поиска.
2. Вероятности событий: производящая функция может быть использована для вычисления вероятностей различных событий, связанных с графами. Например, можно использовать производящую функцию для определения вероятности наличия определенной структуры в случайно сгенерированном графе. Это может помочь оценить вероятности успеха или неудачи алгоритма в зависимости от свойств графа.
3. Время работы алгоритмов: производящая функция может быть использована для анализа времени работы алгоритма на графе. Зная производящую функцию для структуры графа, можно

вычислить ожидаемое количество операций, необходимых для обработки графа с использованием данного алгоритма. Это может помочь оценить эффективность алгоритма и сравнить его с другими подходами.

4. Оптимизация задач: производящая функция может быть использована для оптимизации различных задач на графах. Например, можно использовать производящую функцию для подсчета количества определенных структур в графе и затем установить оптимальные значения параметров алгоритма на основе этих данных. Это может помочь найти наилучшие решения и сократить время выполнения алгоритма.

В целом, анализ алгоритмов на графах с помощью производящих функций позволяет изучать комбинаторные характеристики графов и использовать их для оптимизации и анализа алгоритмических решений. Это может быть полезным в различных областях, таких как анализ данных, оптимизация маршрутов, базы данных, основанные на графах и другие области, где графы играют важную роль.

4. ПРАКТИЧЕСКАЯ ЧАСТЬ. ПОСТРОЕНИЕ ПРОИЗВОДЯЩИХ ФУНКЦИЙ ДЛЯ КОНКРЕТНЫХ ПРИМЕРОВ ГРАФОВ.

Задача: построить производящую функцию для неориентированного графа по его матрице смежности.

Алгоритм программы:

1. Обозначим A как матрицу смежности графа.
2. Вычислим производящую функцию для графа, используя формулу:

$$G(x) = E + A * x + A^2 * x^{\frac{2}{2!}} + A^3 * x^{\frac{3}{3!}} + \cdots$$

   где E - единичная матрица, A^k - матрица, полученная умножением матрицы A саму на себя k раз.

3. Полученная функция G(x) будет являться производящей функцией для графа.

Листинг программы:

```
#include <iostream>
#include <vector>

// Функция для умножения матриц
std::vector<std::vector<int>>
matrixMultiply(std::vector<std::vector<int>>& A,
std::vector<std::vector<int>>& B) {
    int n = A.size();
    std::vector<std::vector<int>> result(n, std::vector<int>(n,
0));

    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            for (int k = 0; k < n; ++k) {
                result[i][j] += A[i][k] * B[k][j];
            }
        }
    }

    return result;
}

// Функция для вычисления факториала числа
int factorial(int n) {
    if (n == 0) return 1;
    return n * factorial(n - 1);
}

int main() {
```

```
    setlocale(LC_ALL, "Russian");
    int n;
    std::cout << "Введите количество вершин в графе: ";
    std::cin >> n;

    std::vector<std::vector<int>> A(n, std::vector<int>(n));
    std::cout << "Введите матрицу смежности графа размером " <<
n << " x " << n << ":\n";
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            std::cin >> A[i][j];
        }
    }

    std::vector<std::vector<int>> I(n, std::vector<int>(n, 0));
    for (int i = 0; i < n; ++i) {
        I[i][i] = 1;
    }

    std::vector<std::vector<int>> G(n, std::vector<int>(n, 0));
    std::vector<std::vector<int>> Ak = I;
    double factorial_val;
    std::cout << "G(x) = ";
    for (int k = 0; k < n; ++k) {
        factorial_val = factorial(k);
        if (k > 0) {
            std::cout << " + ";
        }
        std::cout << "(";
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < n; ++j) {
                if (Ak[i][j] != 0) {
                    if (Ak[i][j] > 1) {
                        std::cout << Ak[i][j] << "*";
                    }
                    std::cout << "x^" << k;
                    if (factorial_val > 1) {
                        std::cout << "/" << factorial_val;
                    }
                    std::cout << " + ";
                }
            }

        }
        std::cout << 0;
        std::cout << ")";
        Ak = matrixMultiply(Ak, A);
    }

    return 0;
}
```

Пример использования:

```
Введите количество вершин в графе: 3
Введите матрицу смежности графа размером 3 x 3:
0 1 1
1 0 1
1 1 0
G(x) = (x^0 + x^0 + x^0 + 0) + (x^1 + x^1 + x^1 + x^1 + x^1 +
x^1 + 0) + (2*x^2/2 + x^2/2 + x^2/2 + x^2/2 + 2*x^2/2 + x^2/2 +
x^2/2 + x^2/2 + 2*x^2/2 + 0)
```

5. ПРАКТИЧЕСКОЕ ЗНАЧЕНИЕ ПРОИЗВОДЯЩИХ ФУНКЦИЙ

Производящая функция является важным инструментом в теории графов, который имеет практическое применение в решении различных задач. Вот некоторые практические примеры использования производящих функций в теории графов:

1. Подсчет числа ребер и вершин: производящая функция может быть использована для определения общего числа ребер и вершин в графе. Различные коэффициенты при переменных в производящей функции представляют количество графов с определенным числом ребер и вершин, что делает этот метод полезным для подсчета этих параметров.
2. Анализ сетей: производящая функция может быть использована для анализа различных параметров сетей, таких как распределение степеней вершин, длина путей, коэффициент кластеризации и т. д. Это может помочь понять свойства и поведение сетей, что в свою очередь может быть полезно в различных приложениях, например, в сетевом анализе социальных сетей или транспортных сетей.
3. Подсчет подграфов: производящая функция может использоваться для подсчета количества подграфов определенного типа или с определенными свойствами. Например, можно подсчитать количество деревьев или циклических подграфов в графе, что может быть полезным при анализе структуры графа или при решении оптимизационных задач.
4. Число возможных путей: производящая функция может использоваться для нахождения числа возможных путей между двумя вершинами в графе. Это может быть полезно, например, при оптимизации маршрутов в транспортных сетях,

моделировании сообщений в сети передачи данных или анализе коммуникационных сетей.

В целом, применение производящей функции в теории графов позволяет получить информацию о структуре графа и его свойствах, что может быть полезным в различных областях, включая компьютерные науки, транспортные системы, социальные сети и другие области, где графы широко используются для моделирования и анализа различных ситуаций и процессов.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы была изучена теория производящих функций и их применение к анализу графов. Были рассмотрены основные понятия производящих функций, их виды для различных типов графов, свойства и способы применения при работе с графами.

Экспериментальное исследование показало, что применение производящих функций к анализу графов позволяет эффективно решать разнообразные задачи, связанные с комбинаторным анализом структурных объектов. Применение данного подхода позволяет упростить вычисления и получить более точные результаты.

Таким образом, выполнение данной курсовой работы позволило углубить знания в области производящих функций и их применение к анализу графов, а также получить практические навыки в решении задач комбинаторики с использованием данного метода.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Wikipedia: сайт. – 2023. – URL: https://ru.wikipedia.org/wiki/Производящая_функция_последовательности (дата обращения: 13.12.2023)
2. С.А. Ландо. Лекции о производящих функциях. – 3-е изд., испр. – Москва: МЦНМО, 2007. - 144 с.
3. StudFIles: сайт. – 2023. – URL: https://studfile.net/preview/2497409/page:5/ (дата обращения: 11.12.2023)
4. Домнин П.Н. Элементы теории графов. – Пенза: Изд-во ПГУ, 2004. –139 с.