Таким образом, в APP Inventor механизм *«Любые компонентные блоки»* реализует инкапсуляцию логики в некоторой конструкции программирования. Такая конструкция заменяет дублированный код для нескольких компонентов, что является эффективным по многим параметрам, в том числе и при необходимости внесения изменения в код программы.

УДК 004.4:519.17

Л. А. ЯРМОЛИК, В. А. ШЕИНА, А. И. ЖУК, Е. Н. ЗАЩУК
Беларусь, Брест, БрГТУ

ОСОБЕННОСТИ ИЗУЧЕНИЯ ТЕОРИИ ГРАФОВ В СИСТЕМЕ WOLFRAM MATHEMATICA

Целью настоящей работы является практическая реализация теории графов в одной из универсальных математических систем Wolfram Mathematica. Рассматриваются способы представления и задания графов, встроенные функции для работы с графами и их тестирования, способы нахождения кратчайшего пути [1]. Приведены рекомендации по использованию Mathematica при обучении теории графов [2, 3].

Визуально граф может быть представлен в виде конечного множества вершин, которые могут быть изображены точкой, и конечного множества их парных связей – линий, соединяющих соответствующие вершины (ребра). В пакете предусмотрены встроенные функции Graph и RandomGraph, которые позволяют изменять цвет, форму и размер вершин, цвет и стиль линии ребер, указывать вес ребер и отображать его на графе [2].

Например, на рисунке 1 сформирован следующий граф из шести вершин, при наведении курсора на который появляется информационное табло, которое позволяет без непосредственного введения встроенных функций вывести информацию о графе на экран. Например, проверить, является ли граф связным или гамильтоновым, содержит ли петли и циклы, вывести степени входа и выхода вершин и список ребер, матрицу смежности и инцидентности и др.

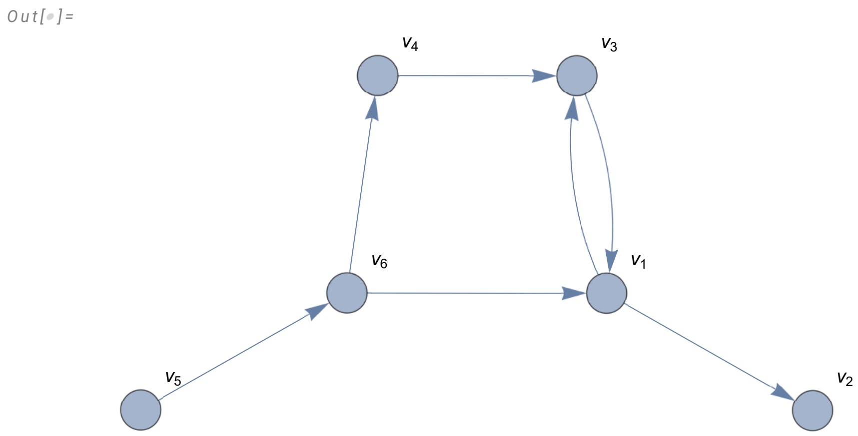

Рисунок 1 – Сгенерированный граф из шести вершин

Рассмотрим задачу о Кенигсбергских мостах: имеются четыре части суши и семь мостов. Необходимо обойти все части суши, пройдя по каждому мосту ровно один раз, и вернуться в исходную точку. Для решения построим граф кенигсбергских мостов с помощью матрицы смежности, используя функцию AdjacencyGraph. На языке графов задача формируется следующим образом: существует ли в графе простой цикл, содержащий все ребра графа (эйлеров цикл)? Для ответа на этот вопрос воспользуемся встроенной функцией FindEulerianCycle.

G1=AdjacencyGraph[0, 1, 1, 1, 1, 0, 2, 0, 1, 2, 0, 2, 1, 0, 2, 0, VertexLabels -> Table[i -> Subscript[v, i], i, 4]]

FindEulerianCycle[G1]

На рисунке 2 показан пример графа, который не является эйлеровым. Следовательно, нельзя обойти все части суши, пройдя по каждому мосту ровно один раз.

Рассмотрим следующую прикладную задачу. Пусть вершины графа – населенные пункты, а связывающие их грунтовые дороги – его ребра, расстояния между населенными пунктами – вес соответствующего ребра. Требуется спланировать наиболее экономичную сеть дорог с твердым покрытием, заменяющих часть грунтовых дорог и связывающую все населенные пункты. Решение этой задачи сводится к построению минимального остовного дерева. Для поиска минимального остовного дерева в системе Mathematica воспользуемся функцией FindSpanningTree и выделим его в исходном графе, используя функцию HighlightGraph, и найдем длину полученной дорожной сети.

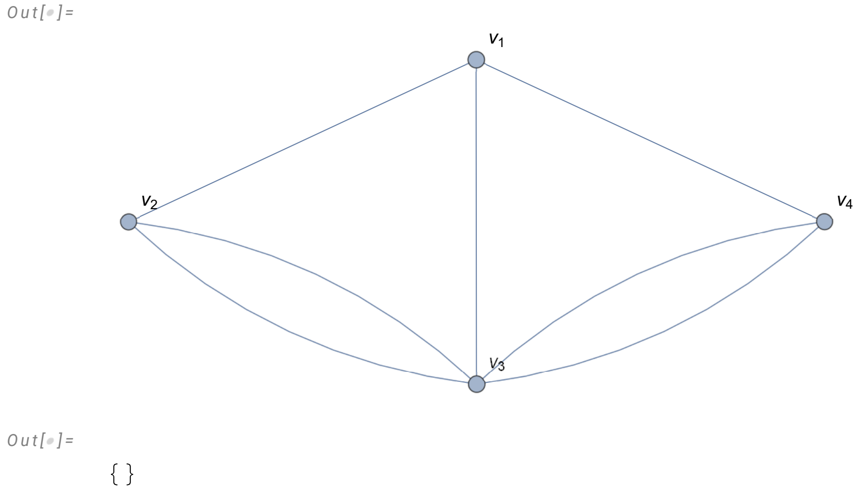

Рисунок 2 – Граф, сформированный по заданной матрице смежности

На рисунке 3 построен минимальный путь по ребрам с указанными весами и найдена его длина, равная 13.

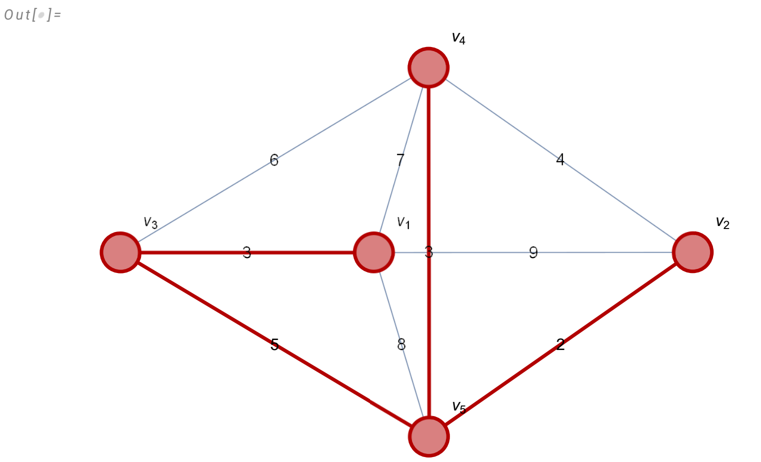

Рисунок 3 – Построение минимального остовного дерева

Следует отметить, что при использовании встроенной функции FindSpanningTree метод поиска выбирается автоматически и зависит от типа и вида графа; если необходимо выбрать метод поиска вручную, можно использовать следующую опцию:
FindSpanningTree[g,Method?"Prim"("Kruskal MinimumCostArborescence)] и задать нужный алгоритм. При решении некоторых прикладных задач существует необходимость найти максимальное остовное дерево, в этом случае также применяется встроенная функция FindSpanningTree, только перед ее применением вес каждого ребра умножается на -1. Построим для того же графа, что и на рисунке 3, максимальное остовное дерево (рисунок 4).

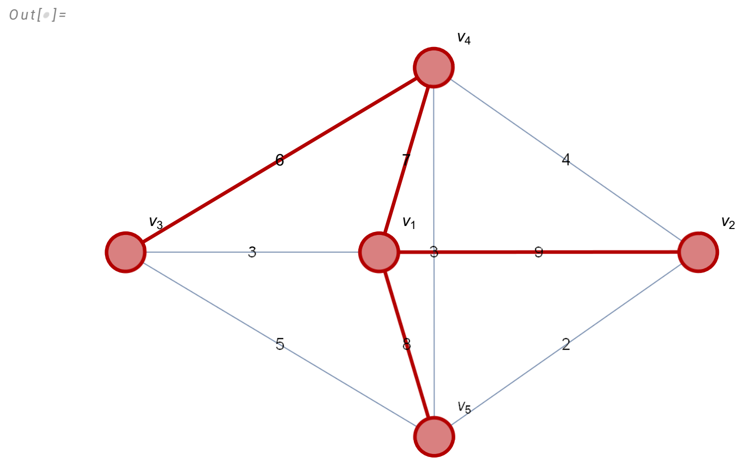

Рисунок 4 – Построение максимального остовного дерева

В ходе проведения исследования были изучены основные встроенные функции в рассматриваемом пакете Mathematica для работы с графами и решены некоторые классические задачи элементов теории графов. В перспективе на основании проведенного исследования можно составить методическую разработку для проведения со студентами лабораторных работ по соответствующей дисциплине, что обеспечит более углубленное и осознанное ее понимание.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Лекции по теории графов / В. А. Емеличев, О. И. Мельников, В. И. Сарванов, Р. И. Тышкевич. – М. : Наука, 1990. – 390 с.

2. Wolfram Demonstrations Project. – URL: https://demonstrations.wolfram.com (дата обращения: 05.04.2025).

3. Защук, Е. Н. Использование методов компьютерной алгебры в лекциях «Предел числовой последовательности и функции» / Е. Н. Защук, А. И. Жук, Л. П. Махнист // Вестник Брестского государственного технического университета. – 2023. – № 1 (130). – С. 125–128.