РОЗДІЛ 9.
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ В ОСВІТІ

ПРОСТЕЙШИЕ ГРАФЫ С MAPLE В НАЧАЛЬНОЙ ШКОЛЕ

SIMPLE GRAPHY WITH MAPLE IN PRIMARY SCHOOL

УДК 373.3

Изворска Д.,
Славова С.

В статье проанализировано использование элементов теории графов, моделированных в системе компьтерной мятематики Maple, при обучении математике в начальной школе. Отличаясь простотой теоретических сведений, наглядностью и доступностью, теория графов может с пользой найти отражение на самом раннем этапе обучения школьников. С помощью этой теории можно решить на доступном для младших школьников уровне ряд достаточно сложных задач во внеклассных занятиях по математике. Использование специализированной библиотеки Networks в Maple, составленная из операторов для работы с графами, позволяет задавать изображения графов и находить их характеристики.

Ключевые слова: *графы, системе компьютерной мятематики Maple, начальная школа.*

У статті проаналізовано використання елементів теорії графів, які смодельвані в системі комп'ютерної математики Maple, при навчанні математики в початковій школі. Відрізняючись простотою теоретичних відомостей, наочністю та доступністю, теорія графів може з користю знайти відображення на самому ранньому етапі навчання школярів. За допомогою цієї теорії можно розв'язати на доступному молодшим школярам рівні ряд досить складних задач на позакласних заняттях з математики. Використання спеціалізованої бібліотеки networks в Maple, яка складена з операторів для роботи з графами, дозволяє задавати зображення графів и знаходити їх характеристики.

Ключові слова: *графи, системи комп'ютерної математики Maple, початкова школа.*

The paper analyzes the use of elements of graph theory modeled in the Maple computer mathematics system when teaching mathematics in elementary school. Featuring simplicity of theoretical information, visibility and accessibility, graph theory can find application at the earliest stage of school children's education. With the help of this theory, it is possible to solve at an accessible for junior pupils level a number of fairly complex tasks in extracurricular classes. The use of the specialized Networks package in Maple made up of operators for working with graphs allows for specifying graph images and finding their characteristics.

Key words: *graphs, system of computer mathematics Maple, primary school.*

Постановка проблемы. Теория графов, наряду с математическим моделированием, является в настоящее время одним из интенсивно развивающихся направлений современной математики, которое связано с теорией множеств и математической логикой, геометрией и топологией, теорией вероятностей и математической статистикой, теорией матриц и другими. Она широко применяется в различных областях науки и техники. При помощи этой теории можно решать большой круг разнообразных математических задач, условия задач приобретают наглядность, их решения являются простыми, без утомительных вычислений, что нельзя не учесть при подготовке будущих начальных учителей. Студенты знакомятся с основными ключевыми понятиями теории графов и их свойствами, решают и составляют задачи, представленные в занимательной форме. Задачи на графы позволяют активно использовать наглядное изображение графа для поиска решения. Долгое время задачи теории графов решались вручную, но с появлением компьютера появилась возможность использовать его как средства решения научных и прикладных задач. Графическое представление можно получить с помощью компьютерных программ обработки графов, которые позволяют легко редактировать изображение графа.

Постановка задачи (целей статьи). Первые программы обработки математических данных на компьютерах – системы компьютерной математики (СКМ) появились в 80-х годах прошлого столетия. В настоящее время они имеют высочайший математический уровень по объему, реализованных в них математических методов и по самим реализациям. Их обновления продолжаются, каждая следующая версия расширяет возможности предыдущей, но основы не меняются. Среди известных и широко используемых СКМ следует отметить *Maple, Mathematica, Mathcad, Matlab, Reduce, Derive.* Наличие развитых СКМ требует значительной переориентации учебного процесса. Преподавание многих учебных курсов может стать значительно эффективнее, если активно использовать их графические возможности и автоматическое проведение трудоемких алгебраических выкладок. СКМ позволяют упрощать алгебраические выражения, выражать неизвестные из уравнений и систем уравнений, решать неравенства и системы неравенств, строить графики и многое другое. Для решения задач, объектами которых являются графы, эти пакеты незаменимы. СКМ в значительной мере расширяют круг дидактических средств обучения. На практических и лабораторных занятиях студенты знакомятся с использованием таких программ и применяют их в процессе поиска решения или проверки найденного решения задачи.

Внеклассные занятия по математике для учащихся III-IV классов направлены на повышение уровня логического и математического развития учащихся, формирование у них математической грамотности и абстрактного мышления, подготовка к восприятию понятий математики и ее методов в следующих классах. Мышление учеников является в основном конкретно-ситуационным, с V класса начинается постепенный переход к абстрактному, оперирующему отвлеченными понятиями, суждениями, умозаключениями, мышлению. Можно выделить следующие основные задачи внеклассных занятий: 1) ознакомить учащихся с простейшими способами рассуждений при помощи неявного использования мыслительных операций; с приемами решения логических задач; с понятием доказательства в математике; с простейшими способами построения математических моделей; 2) научить учащихся пользоваться вышесказанным для решения логических и математических задач. Изучение материала проходит блоками через систему тематически ориентированных задач. Содержание обучения для III класса повторяется в IV классе на более сложном уровне. Большинство задач должно быть занимательными. Такие задачи повышают интерес школьников к занятиям математикой, позволяют использовать в процессе обучения их любознательность. В качестве одного из средств решения задач используется построение математических моделей описываемых ситуаций.

Знакомство с некоторыми понятиями теории граф и их свойствами становится возможным в начальной школе при решении логических и комбинаторных задач, задач на начертание фигур одним росчерком, последовательности чисел и др. [1]. С графами мы сталкиваемся постоянно, например, графом является схема линий метрополитена. Исследуя свою родословную, строим так называемое генеалогическое древо – граф. Графы служат удобным средством описания связей между объектами. Понятие графа целесообразно вводить после нескольких задач, решающее соображение в которых графическое представление. Важно понять, что один и тот же граф может быть нарисован разными способами. Строгое определение графа давать не нужно, важно, чтобы ученики разобрались на примерах и различали его виды (неориентированные, ориентированные, полные, деревья, эйлеровы). Содержательным свойством является утверждение о четности числа нечетных вершин. Главная цель, которую нужно преследовать при изучении графов, – научить школьников видеть граф в условии задачи и грамотно переводить условие на язык теории графов. Граф используют не только как иллюстрацию, например, рассматривая граф, изображающий сеть дорог между населенными пунктами, можно определить маршрут проезда от пункта А до пункта Б. Если таких маршрутов несколько, можно выбрать в определенном смысле оптимальный, например самый короткий или самый безопасный.

Впервые понятие «граф» ввел в 1936 году венгерский математик *Денеш Кёниг,* назвав графами схемы, состоящие из множества точек и связывающих эти точки отрезков прямых и кривых линий [3]. Однако первая работа по теории графов была написана еще в 1736 году швейцарским и российским математиком *Леонардом Эйлером* (1707–1783), в которой он решил «задачу о Кёнигсбергских мостах», ставшую одной из классических задач теории графов. На рис. 1, слева представлен схематический план центральной части города Кенигсберг, включающий два берега реки Перголя, два острова в ней и 7 соединяющих мостов. Задача состоит в том, чтобы обойти все четыре части суши, пройдя по каждому мосту один раз, и вернуться в исходную точку [3]. Эйлер «сжал» сушу в точки, мосты «вытянул» в линии и в результате получилась фигура на рис. 1, справа. Он показал, что нельзя обойти семь городских мостов и вернуться в исходную точку, пройдя по каждому мосту ровно один раз, и придумал общий метод решения подобных задач. До конца XIX века графы применялись лишь при решении занимательных задач. В начале XX века теория графов оформилась как самостоятельная математическая дисциплина.

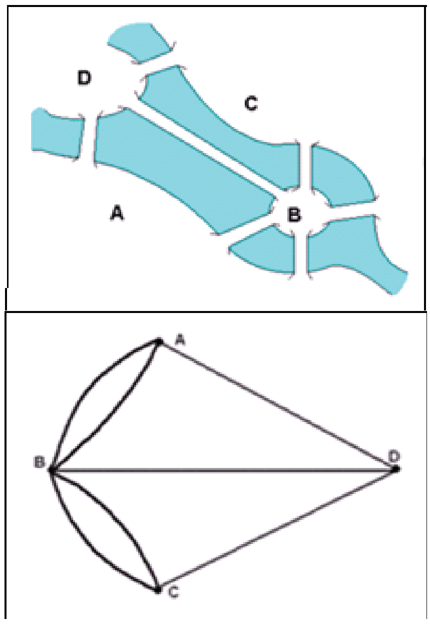

Рис. 1. Задача о Кенигсбергских мостах

Такую фигуру, состоящую из точек и линий, связывающих эти точки, называют *графом.* Точки А, В, С, D называют *вершинами* графа, а линии, соединяющие их – *ребрами* графа. У графа обязательно есть вершины. Вершины графа, которые не принадлежат ни одному ребру, называются *изолированными*. Граф, состоящий только из изолированных вершин, называется *нуль-графом (пустым).* Граф, в котором каждая пара вершин соединена ребром, называется *полным* (рис. 2). K_n – граф, состоящий из *n* вершин и ребер, соединяющих всевозможные пары этих вершин. Такой граф можно представить как *n*-угольник, в котором проведены все диагонали. Графы, в которых не построены все возможные рёбра называются *неполными.*

Дополнением графа G называется граф с теми же вершинами, что и граф G, и с теми и только теми ребрами, которые необходимо добавить к графу, чтобы получился полный граф. Если по рёбрам можно перемещаться в обоих направлениях, то граф называется неориентированным. Если по каждому ребру можно пройти только в одну сторону, то граф ориентированный. В таком случае рёбра обычно обозначаются стрелками, а не просто линиями.

Вершины, из которых выходит нечетное число ребер, называют *нечетными*, а вершины, из которых выходит четное количество ребер, – *четными*. Решая задачу про кенигсбергские мосты, Эйлер установил, что если все вершины графа четные, то можно одним росчерком (т.е. не отрывая карандаша от бумаги и не проводя дважды по одной и той же линии) начертить граф. Можно начать с любой вершины и окончить в той же вершине. Граф с двумя нечетными вершинами также можно начертить одним росчерком, начиная движение с одной и заканчивая на другой нечетной вершине. Граф с более чем двумя нечетными вершинами невозможно начертить одним росчерком. В задаче о семи кенигсбергских мостах все четыре вершины графа нечетные, т.е. невозможно пройти по всем мостам один раз и закончить путь там, где был начат.

Степенью вершины называется количество выходящих из нее ребер. Следовательно, вершина – нечетная (четная), если ее степень – число нечетное (четное). Обозначение: p (A) – степень вершины A. На рис.1, справа из вершин A, C и D выходят по 3 ребра, а из вершины B – 5 ребер, поэтому p(A) = p(C) = p(D) = 3 и p(B) = 5. Граф, степени всех *k* вершин которого одинаковы, называется *однородным графом степени k*. С понятием степени вершины связаны две *основные теоремы* теории графов: 1) сумма степеней вершин любого графа равна удвоенному числу его ребер; 2) любой граф содержит четное число нечетных вершин.

Система *Maple,* продукт канадской компании Waterloo Maple Inc [4], является очень хорошим средством интенсификации учебного процесса при подготовке учеников начальной школы. В момент создания (*Maple* 1.0: январь 1982) система предназначалась как для общих научных целей, так и для целей обучения. Она позволяет решать различные математические задачи, имеет хорошие графические возможности. Многие стандартные задачи, требующие большого количества несложных выкладок, в системе *Maple* выполняются в доли секунды. Наличие в системе своих средств программирования позволяет реализовать в ее рамках и простейшие геометрические алгоритмы. Последние версии системы являются *Maple 18* (7 марта 2014) и *Maple 2015* (5 марта 2015). Работа в *Maple* проходит в режиме сессии: пользователь вводит предложения (команды, выражения, процедуры), которые воспринимаются условно и обрабатываются *Maple*. Рабочее поле разделяется на 3 части: 1) область ввода – состоит из командных строк, команды вводятся после оператора >; 2) область вывода – содержит результаты обработки введенных команд в виде аналитических выражений, графических обьектов или сообщений об ошибке; 3) область текстовых комментариев – содержит текстовую информацию, которая может пояснить выполняемые процедуры; текстовые строки не воспринимаются *Maple* и не обрабатываются. Для отмены всех сделанных назначений и начала нового сеанса без выхода из системы используется команда *restart.* В состав *Maple* входит большое число библиотек, подключение которых осуществляется командой *with* (имя библиотеки). Например, библиотека *networks* предназначена для работы с графами.

Оператор >, вставляемый кнопкой > строки инструментов основной панели указывает начало ввода информации, которую надо обработать. Она вводится красным шрифтом, с выравниванием по левому краю. Ввод, начинающийся с оператора >, называется активным, а строка, открываемая этим оператором, – командной строкой. После опера-

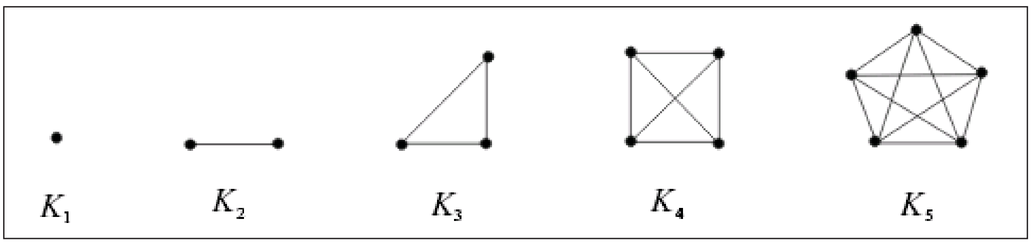

Рис. 2. Полные графы

тора > устанавливается визир курсора в форме вертикальной черточки I и вводится информация. Оператор «Вернуть результат» – точка с запятой <;>. Если командная строка заканчивается им, и нет ошибок ввода, то в каком бы месте строки не находился курсор, по команде <Enter> появляется результат. В командной строке может быть несколько операторов <;>. Сколько их будет, столько будет и результатов. Двоеточие <:> – оператор отказа от вывода результата на рабочий лист.

Использование специализированной библиотеки *networks*, составленная из операторов для работы с графами, позволяет не только задавать изображения графов и находить их характеристики, но и программировать алгоритмы, что дает возможность хоть каким-то образом обратиться к основным алгоритмам теории графов и освоить их при помощи компьютера. Достаточно широкий спектр задач можно решить при помощи теории графов. Команда подключения этой библиотеки – стандартная, т.е. достаточно воспользоваться оператором *with.* Граф в *Maple* представляется особой процедурой типа *GRAPH*. Для работы с графами можно воспользоваться любой из 75-ти функций, содержащихся в библиотеке networks. Основные функции, которые позволяют создавать и изменять графы, а также процедуры решения классических задач из теории графов приведены в таблице 1 [2].

Рассмотрим применение команд *complete, draw, delete, cycle*, *void, connect, complement, vdegree, nops(edges)* и *nops (vertices)* в Maple 10 к решению задач на внеклассных занятиях по математике в III и IV классах..

Задача 1. В шахматном турнире участвуют 4 школьника: Андрей, Борис, Виктор и Дмитрий. Первенство проводится по круговой системе – каждый из участников играет с каждым из остальных один раз. Сколько всего сыграно партий? А если участвует и Галина? А если присоединится и Елена?

Продемонстрируем случай для 6 участников. Задача может быть решена при помощи таблицы (рис. 3, слева) или графа (рис. 3, справа). Ученики заполняют таблицу коллективно и открывают структуру решения в последнем ряду: 1 + 2 + 3 + 4 + 5 = 15. После короткого обсуждения они соображают, что для 5 участников (4 участников) решение получается сразу: 1 + 2 + 3 + 4 = 10 (1 + 2 + 3 = 6). Для решения задачи графично нужно изобразить участников турнира точками, а сыгранные ими партии – отрезками. Получается полный граф с шестью вершинами и надо лишь подсчитать число ребер

Таблица 1

Основные функции библиотеки *networks* [2]

Функции создания графов	
complete	создает полный граф
complement	создает дополнение графа
random	возвращает случайный граф
void	создает пустой граф (без ребер)
Функции модификации графов	
addvertex	добавляет в граф вершины
addedges	добавляет в граф ребро
delete	удаляет из графа ребро или вершину.
connect	соединяет заданные вершины с другими
cycle	соединяет последовательные вершины
Функции контроля структуры графов	
draw	рисует граф
edges	возвращает список ребер графа
vertices	возвращает список вершин графа
ends	возвращает имена вершин графа
head	возвращает имя вершины - голова ребер
tail	возвращает имя вершины - хвост ребер
shortpathtree	нахождение дерева кратчайших расстояний
show	вывод полной информации о графе
Функции вычисления элементов графа	
nops(edges)	определяет число ребер графа
nops (vertices)	определяет число узлов графа
vdegree	определяет степень вершин графа

графа. Из таблицы 1 ученики выбирают команду *complete,* а учитель демонстрирует последовательност команд в *Maple.* Чтобы подсчитать число ребер графа надо перемножить число вершин графа с числом ребер, выходящими из них и полученное произведение разделить на 2, так как каждое ребро учитывается дважды: (5.4):2=10. Потом ученики самостоятельно записывают команды для построения полного графа с 4 вершинами (рис. 4, слева). и с 5 вершинами (рис. 4, справа)) и подсчитают число ребер.

Maple предлагает и другой способ построения графа с помощью команды *draw*, являющимся универсальным для любого графа. Он основан на последовательную запис всех ребер графа. Учитель записывает последовательность команд для графа G3:

> restart: with(networks):

> **G3:=graph({1,2,3,4,5},{{1,2},{1,3},{1,4},{1,5},{2,3},{2,4},{2,5},{3,4},{3,5},{4,5}}):**

> draw(G1);

Потом ученики модифицируют эти команды для графа G2:

> restart: with(networks):

>**G2:=graph({1,2,3,4},{{1,2},{1,3},{1,4},{2,3},{2,4},{3,4}}):**

> draw(G2);

Графы G2 и G3 можно получить из графа G1, построенного командой complete и затем преобразовать его путем удаления части вершин командой *delete*. Исходный и преобразованный графы строятся функцией draw. Учитель записывает последовательность команд для графа G3:

> restart: with(networks):

> G1:=complete(6): draw(G1);

> delete({6},G1): draw(G1);

Ученики модифицируют эти команды для графа G2:

> restart: with(networks):

> G1:=complete(6): draw(G1);

> delete({5,6},G1): draw(G1);

Сравнивая рассмотренные три способа построения полного графа, ученики делают вывод о рациональности первого способа при помощи команды *complete.*

Задача 2. Запишите последовательность команд в Maple, которыми построен граф G4 на рис. 5. Составьте и решите задачу, моделированную этим графом

Ученики самостоятельно определяют, что граф G4 является полным графом с 14 вершинами, что и определяет следующую последовательность команд в Maple

> restart: with(networks):

> G4:=complete(14): draw(G4);

	А	Б	В	Г	Д	Е	Всего
А	0	1	1	1	1	1	5
Б	0	0	1	1	1	1	4
В	0	0	0	1	1	1	3
Г	0	0	0	0	1	1	2
Д	0	0	0	0	0	1	1
Е	0	0	0	0	0	0	0
Всего	0	1	2	3	4	5	15

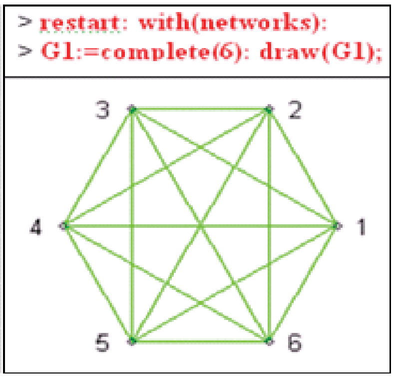

Рис. 3. Моделирование турнира с 6 участниками

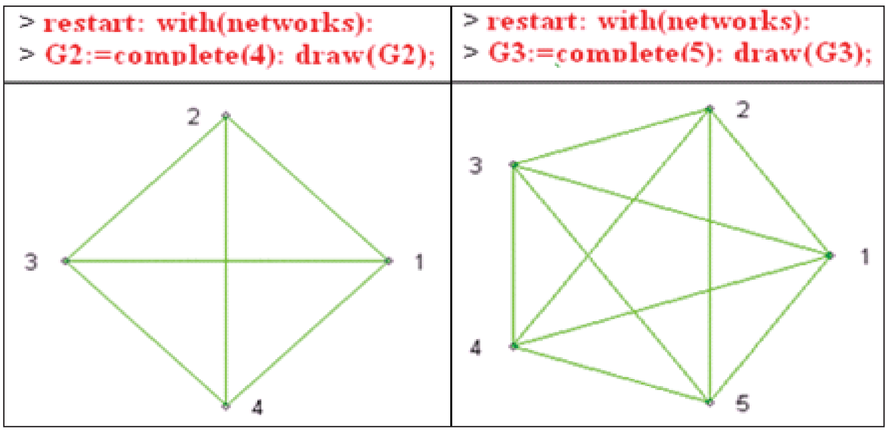

Рис. 4. Моделирование турнира с 4 и 5 участниками

Граф G4 моделирует, например, следующую задачу: Четырнадцать учеников IV класса встретились на вокзале, чтобы поехать за город в лес. При встрече они поздоровались друг с другом за руку. Сколько всего было рукопожатий?

На графе 91 ребер, следовательно, было сделано 91 рукопожатий. Для того чтобы упростить подсчет ребер графа, можно рассуждать так: из каждой из 14 вершин выходят 13 ребер, т.е. для подсчета ребер умножаем 14 на 13, но при этом каждый отрезок мы посчитаем дважды. Следовательно, получаем (14.13):2 = (13.14):2 = =13.(14:2) = 13.7 = 91 ребер графа. В случае мы применили основное свойство любого графа: сумма степеней вершин графа равна удвоенному числу его ребер (поскольку данный граф является полным, то все его вершины имеют степень 10). Такие рассуждения особенно полезны, когда граф содержит большее число вершин и подсчет числа ребер становится затруднительным.

Рассмотрим и обратную задачу.

Задача 3. Несколько учеников IV класса встретились на вокзале, чтобы поехать за город в лес. При встрече они поздоровались друг с другом за руку. Сколько учеников поехало за город, если всего было: а) 55 рукопожатий? б) 40 рукопожатий?

В случай а) было сделано 55 рукопожатий, следовательно на моделирующем графе G5 55 ребер. Если обозначить число вершин графа (число учеников) через n, то n.(n-1):2 = 55, т.е. n.(n-1) = 55.2 или n.(n-1) = 110, или n.(n-1) = 11.10. Следовательно, у графа 11 вершин, т.е. на вокзале было 11 учеников. В случай б) аналогичным образом получаем n.(n-1):2 = 40, т.е. n.(n-1) = 40.2 или n.(n-1) = 80; Задача не имеет решения, так как 80 нельзя представить произведением двух последовательных натуральных чисел. Граф G5 изображен на рис. 6.

```
> restart: with(networks):
> G5:=complete(11): draw(G5);
```

В рассмотренных выше задачах необходимо знать количество ребер, вершин графа и их степень. Посчитать по рисунку иногда невозможно из-за большого количества ребер и вершин. В *Maple* ест команды *nops(edges)* для числа ребер, *nops(vertices)* для числа вершин и *vdegree* для степени вершин. Например, для числа ребер и для степени вершин графа G4 в задаче 2 получаем

```
> nops(edges(G4)); 91
> vdegree(1,G4); 13
```

Поскольку степени вершин равные (почему?), достаточно определить степень одной вершины, например 1. Если перемножить ее на число вершин 14, получим сумму степеней вершин графа (13.14), которая равна удвоенному числу ребер графа...

Для решения следующих задач нужны будут новые команды из таблицы 1.

Задача 4. В соревнование по волейболу участвуют 8 команд. Нарисуйте граф, в котором вершинами являются команды, а ребрами – игры, сыгранные между командами н следующих случаях: а) до начала соревнования; б) если сыграли следующие команды – 1 с 2, 2 с 3, 3 с 4, 4 с 5, 5 с 6, 6 с 7, 7 с 8, 8 с 1; в) после окончания соревнования; г) если команда 8 съграла с другими командами; д) если сыграли следующие команды – 1 с 2, 5, 4; 3 с 2, 7, 4; 8 с 5, 6, 7, 4; 6 с 2, 5, 7, 4; 7 с 2, 5; е). Какие игри не сыграны в случай г) и д) и сколько их?

В случае а) у графа G6 только вершины, он пустой и нужно использовать команду *void* (рис. 7, слева). В случае б) нужно соединить последовательно вершины графа G7 коман-

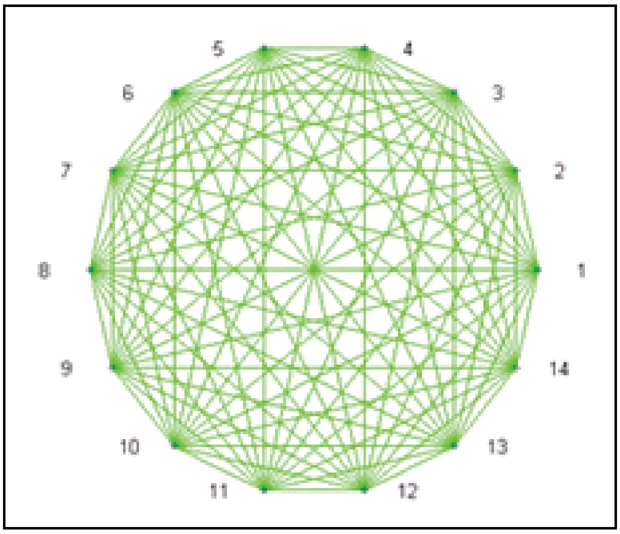

Рис. 5. Граф G4

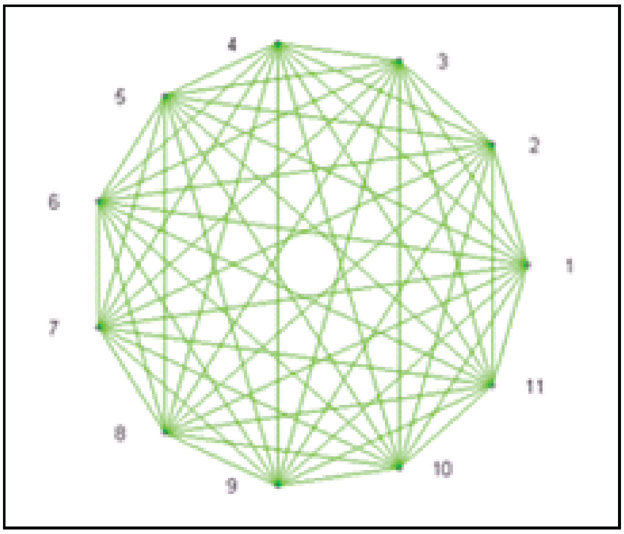

Рис. 6. Полный граф G5

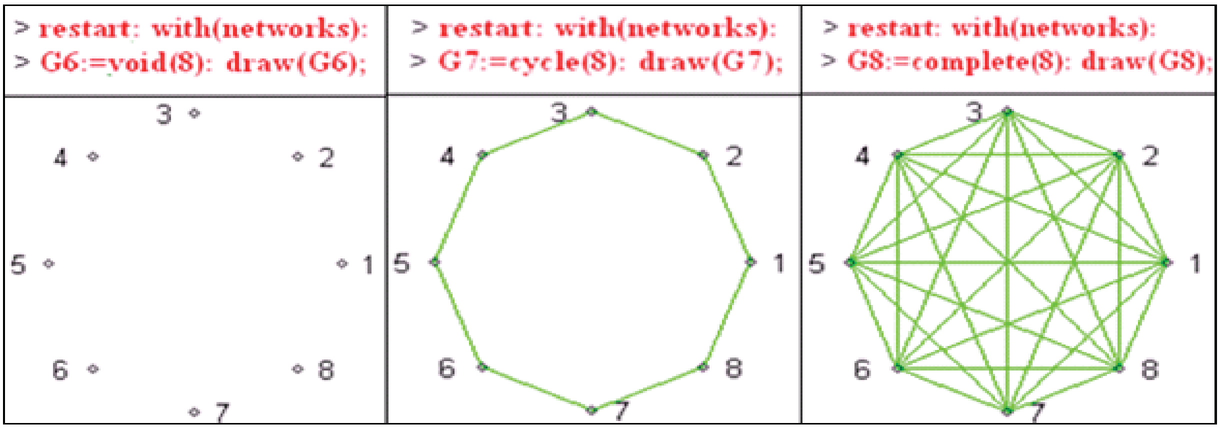

Рис. 7. Графы G6, G7 и G8

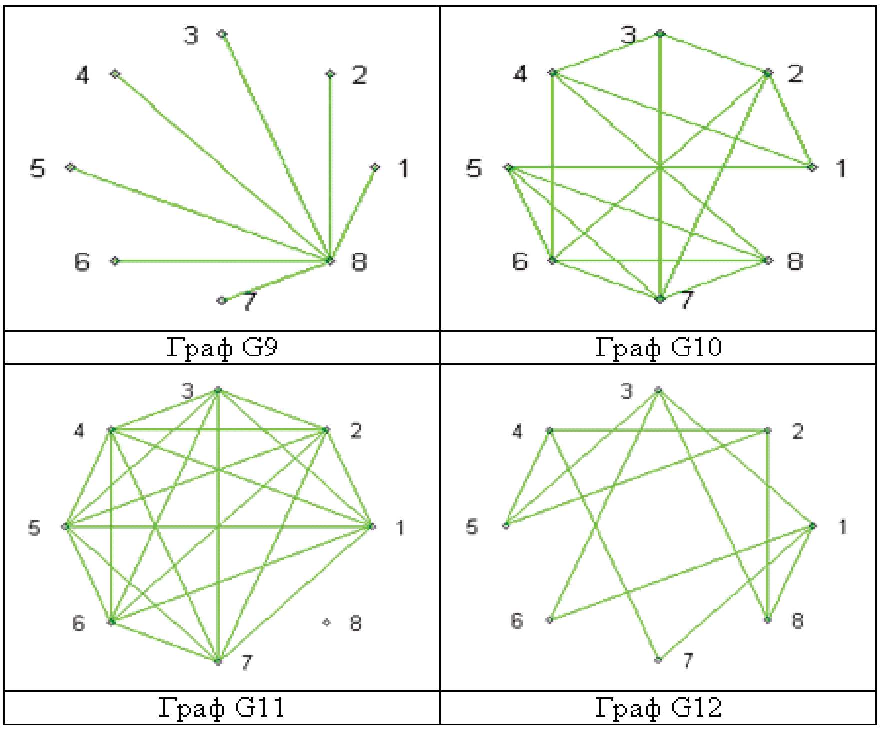

Рис. 8. Графы G9, G10 и их дополнения графы G11, G12

дой *cycle* (рис.7, в середине). В случае в) граф G8 является полным и задается командой *complete* (рис. 7, справа).

В случаях г) для графа G9 и д) для графа G10 сначала создадим вершины графа командой *void,* потом из таблицы 1 выбираем команду *connect*, которая соединяет заданные вершины с другими (рис. 8, первый ряд).

> **restart: with(networks):**

> **G9:=void(8): connect({8},{1,2,3,4,5,6,7},G9): draw(G9);**

> **G10:=void(8): connect({1},{2,5,4},G10): connect({3},{2,7,4},G10): connect({8},{5,6,7,4},G10): connect({6},{2,5,7,4},G10): connect({7},{2,5},G10): draw(G10);**

В случае е) нужно построить графов G11 и G12 при помощи команды *complement* (рис.8, второй ряд)

> **G11:=complement(G9): draw(G11);**

> **G12:=complement(G10): draw(G12);**

Отметим, что вершина 8 является изолированной для графа G11. Чтобы определить число несыгранных игр, нужно определить число ребер в графах G11 и G12 командой *nops(edges):*.

> **nops(edges(G11));** 21

> nops(edges(G12)); 12

Выводы из проведенного исследования. Обучение задачам практического характера и простейшим задачам теории графов с *Maple* является эффективным средством повышения эффективного развития практических навыков учащихся. Создать методику решения задач практического характера и введения элементов теории графов способствовала бы повышению эффективности прикладного и практического аспектов математического курса начальной школы, в частности: а) методику моделирования и разработки ситуаций задач практического характера; б) составление системы задач практического характера как для обязательного курса, так и для внеклассных занятий; в) составление системы задач по теории графов; г) подготовку методической рекомендации для учителей как для обучения теоретическому материалу, так и для работы с *Maple*, решения задач; д) предложение методической рекомендации для введения элементов теории графов и решения соответствующих задач.

БІБЛІОГРАФІЧНИЙ СПИСОК:

1. Аммосова, Н.В., Коваленко, Б.Б. Использование теории графов в математическом образовании школьников, МКО. 2009. Т. 1. С. 123–132.
2. Кирсанов М.Н. Графы в Maple, М.: Физматлит, 2007. 168 с.
3. Татт У. Теория графов. М.: Мир, 1988, 424 с.
4. Farr, J., M. Khatarinejad Fard, S. Khodadad, M. Monagan. A GraphTheory Package for Maple, Proceedings of the 2005 Maple Conference, pp. 260-271, Maplesoft, 2005.