Т.В. Бурзалова
Россия, Улан-Удэ, Бурятский государственный университет

Некоторые задачи теории графов в системе "Mathematica"

В данной работе рассматривается применение компьютерной системы «Mathematica» при решении комбинаторных задач школьной математики. Внедрение этой методики в школьный курс значительно повышает эффективность обучения решению сложных и трудоемких задач комбинаторики.

T.V. Burzalova

Some tasks of graph theory in system " Mathematica "

In the given work is considered application of computer system " Mathematica " at the decision of combinatory tasks of school mathematics. Introduction of this technique in a school rate considerably raises a learning efficiency to the decision of difficult and toilful tasks of graph theory.

Изучение элементов теории графов в школе представляется целесообразным по нескольким причинам. Прежде всего, теория графов является важной областью современной математики, имеющей широкое практическое применение. В настоящее время такие дисциплины, как физика, химия, экономика, исследование операций, социология, а также многие другие, имеющие важное прикладное значение, широко используют методы дискретной математики и полученные в ней результаты. Эти результаты очень важны, например, при разработке различных технических устройств с дискретным принципом действия, построения машинных алгоритмов, в планировании и радиоэлектронике. Кроме того, рассматриваемый материал является наглядным, отличается богатым, но в то же время вполне доступным для понимания учащихся математическим аппаратом. Сторонники включения в систематический курс математики теории графов и ее приложений [1,2] обращают внимание на доступность, наглядность, и, что немаловажно, широкое применение теории графов к решению задач (арифметических, логических, комбинаторных задач на сравнение числа элементов множеств).

В дискретной математике особое место занимают задачи, связанные с упорядочиванием тех или иных объектов и построением сложных конструкций путем «правильного» соединения отдельных элементов, а также задачи, в которых изучаются отношения между различного рода объектами. Это задачи, решение которых сводится к поиску гамильтонова пути или кратчайшего пути на графах.

Приведем решение нескольких таких задач с использованием компьютерной системы "Mathematica". Стандартный пакет расширения <<DiscreteMath`Combinatorica` этой компьютерной системы обладает богатейшим набором встроенных функций теории графов и комбинаторики (около 450). Он содержит функции для конструирования графов и других комбинаторных объектов, вычисляет их инварианты и, наконец, позволяет создать их графическую визуализацию.

Задача. Три ревнивых мужа и их жены должны переправиться через реку. Имеется только один маленький бот, который может выдержать одновременно только двоих человек. Как могут переправиться все шестеро, если никакой муж не оставит жену в присутствии других мужчин?

Решение. Пусть нужно переправиться с левого берега на правый берег. Сформируем множество l всех возможных сочетаний людей на левом берегу, причем в каждом сочетании последний символ b означает наличие бота у группы оставшихся, а символ o – отсутствие такового.

```
In[2]:= l = {{m1, m2, m3, v1, v2, v3, b}, {m1, m2, m3, v1, b}, {m1, m2, m3, v2, b}, {m1, m2, m3, v3, b},
    {m1, m2, v1, v2, b}, {m1, m3, v1, v3, b}, {m2, m3, v2, v3, b}, {m1, m2, m3, v2, b},
    {m1, m2, m3, v3, b}, {m1, m2, m3, v2, v3, b}, {m1, m2, m3, v1, v3, b}, {m1, m2, m3, v1, v2, b},
    {m1, v1, b}, {m2, v2, b}, {m3, v3, b}, {v1, v2, v3, b}, {v1, v2, b}, {v1, v3, b}, {v2, v3, b},
    {m1, m2, m3, v1, o}, {m1, m2, m3, v2, o}, {m1, m2, m3, v3, o}, {m1, m2, v1, v2, o},
    {m1, m3, v1, v3, o}, {m2, m3, v2, v3, o}, {m1, m2, m3, v2, o}, {m1, m2, m3, v3, o}, {m1, m2, m3, o},
    {m1, v1, o}, {m2, v2, o}, {m3, v3, o}, {v1, v2, o}, {v1, v3, o}, {v2, v3, o}, {v1, o}, {v2, o}, {v3, o}, {o}};
```

Этот список содержит 38 элементов.

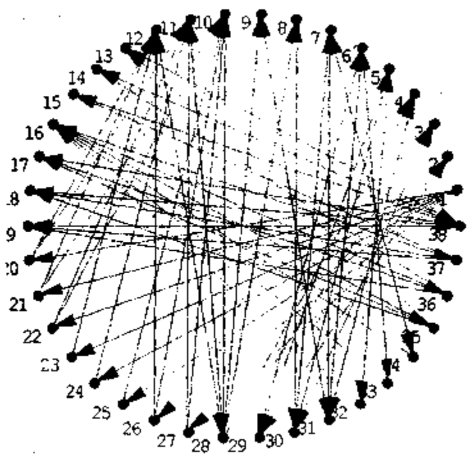
```
In[3]:= Length[l]

Out[3]= 38
```

Построим бинарное отношение на множестве l:

```
In[4]:= kr =
    Module[{t = Complement[#1, #2]},
      Length[#2] - Length[#1] == 1 && Last[#1] =!= Last[#2] && Last[#1] == o && Length[#2] =!= 7 &&
        Length[Complement[#2, #1]] == 2 || Length[#2] - Length[#1] == 2 && Last[#1] =!= Last[#2] &&
        Last[#1] == o && Length[#2] =!= 7 && Length[Complement[#2, #1]] == 3 ||
      (Length[#1] - Length[#2] == 2 && Last[#1] =!= Last[#2] && Length[Complement[#1, #2]] == 3 && Last[#1] == b)] &;
```

Построим граф с вершинами в точках множества l, причем две вершины связаны ребром, если они удовлетворяют отношению kr:

```
In[5]:= ShowLabeledGraph[g = MakeGraph[l, kr], EdgeDirection -> True]
```



Рассмотрим кратчайший путь из первой вершины в тридцать восьмую:

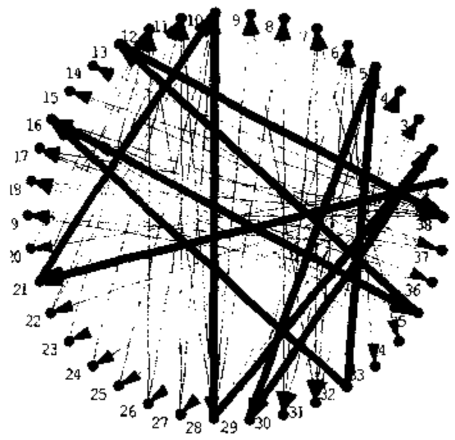
```
In[6]:= ShortestPath[g, 1, 38]

Out[6]= {1, 21, 10, 28, 2, 29, 5, 32, 16, 35, 13, 38}
```

Рассмотрим исходные метки соответствующих вершин, именно они показывают искомое решение задачи:

```
In[7]:= Map[l[[#]] &, ShortestPath[g, 1, 38]]

Out[7]= {{m1, m2, m3, v1, v2, v3, b}, {m1, m2, m3, v2, o}, {m1, m2, m3, v2, v3, b}, {m1, m2, m3, o},
    {m1, m2, m3, v1, b}, {m1, v1, o}, {m1, m2, v1, v2, b}, {v1, v2, o}, {v1, v2, v3, b}, {v1, o}, {m1, v1, b}, {o}}
```

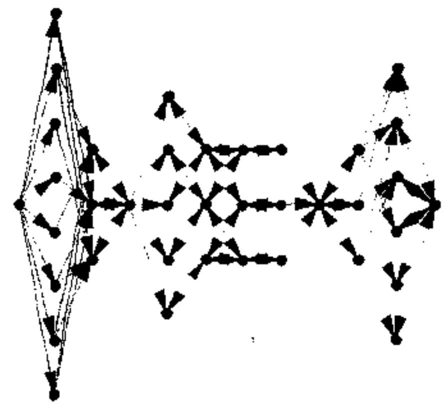
Выделим кратчайший путь от первой до тридцать восьмой вершины:

```
In[8]:= ShowLabeledGraph[Highlight[g, {Partition[ShortestPath[g, 1, 38], 2, 1]}]]
```



Улучшим вложение графа с помощью встроенной функции RankedEmbedding:

```
In[9]:= ShowGraph[RankedEmbedding[g, {1}], EdgeDirection → True]
```



Выделим гамильтонов цикл:

```
In[10]:= ShowGraph[Highlight[RankedEmbedding[g, {1}], {Partition[ShortestPath[g, 1, 38], 2, 1]}]]
```

Задача. Требуется перечислить все перестановки множества $\{1,2,\ldots,n\}$ в таком порядке, чтобы каждая последующая перестановка отличалась от предыдущей ровно одной транспозицией.

Решение. Достаточно свести задачу к нахождению гамильтонова пути на графе.

На множестве всех перестановок определим бинарное отношение P: две перестановки находятся в отношении P, если они отличаются друг от друга ровно одной транспозицией. Как известно, бинарное отношение можно представить графом, вершины которого соответствуют элементам множества, и две вершины смежны, если они находятся в данном отношении.

Заметим, что две перестановки находятся в отношении P, если они отличаются как упорядоченные множества ровно в двух позициях. Следующая функция определяет P:

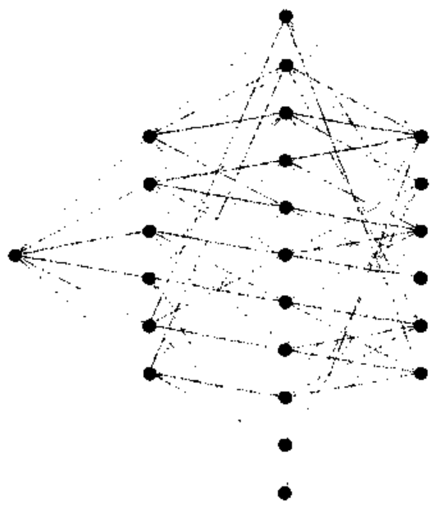
```
In[25]:= P = (p = #1; q = #2; k = 0; Do[If[p[[i]] == q[[i]], , k = k + 1], {i, 1, Length[#1]}];
        If[k == 2, True, False]) &;
```

Мы определили функцию P от двух аргументов, здесь p=#1 –первая перестановка, q=#2- вторая перестановка, далее в цикле Do[] подсчитываем количество k несовпадающих позиций в этих перестановках. Функция возвращает True, если k=2.
Рассмотрим пример для перестановок длины 4.

```
In[27]:= g = MakeGraph[Permutations[4], P, Type → Undirected]

Out[27]= -Graph:<72, 24, Undirected>-
```

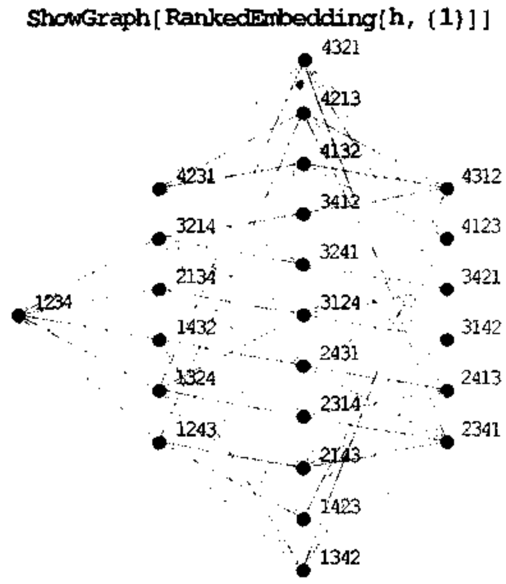
Для наглядности построим ранжированное вложение вершин этого графа относительно первой вершины:

```
In[28]:= ShowGraph[RankedEmbedding[g, {1}]]
```



Если мы хотим определить перестановку, соответствующую каждой вершине графа g , то достаточно в теле графа установить опцию VertexLabel->True, но метки сильно загромождают рисунок.

```
In[29]:= h = MakeGraph[Permutations[4], P, Type → Undirected, VertexLabel → True];
     ShowGraph[RankedEmbedding[h, {1}]]
```

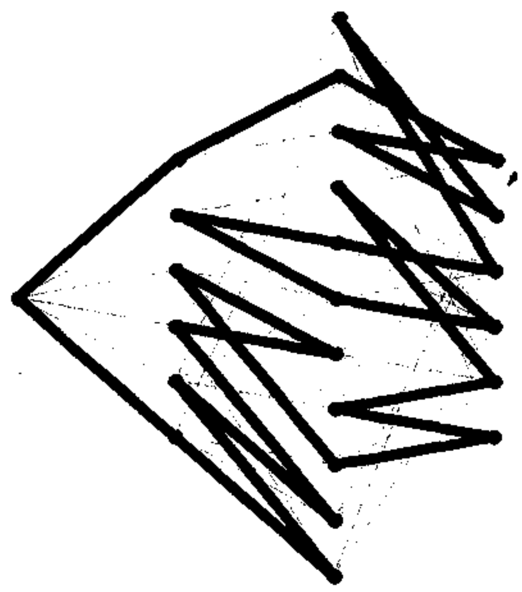


Проверим, является ли граф g гамильтоновым:

```
In[30]:= HamiltonianCycle[g]

Out[30]= {1, 2, 4, 3, 5, 6, 12, 7, 8, 10, 9, 11, 17, 14, 13, 15, 16, 18, 24, 19, 20, 23, 21, 22, 1}
```

Выделим этот гамильтонов цикл в графе:

```
In[31]:= ShowGraph[RankedEmbedding[Highlight[g, {Partition[HamiltonianCycle[g], 2, 1]}], {1}]]
```



Пройдя по вершинам этого гамильтонова цикла, получим все перестановки размера 4, причем любые две смежные перестановки отличаются ровно одной транспозицией.

```
In[32]:= GetVertexLabels[h, HamiltonianCycle[h]]

Out[32]= {1234, 1243, 1342, 1324, 1423, 1432, 2431, 2134, 2143, 2341, 2314,
    2413, 3412, 3142, 3124, 3214, 3241, 3421, 4321, 4123, 4132, 4312, 4213, 4231, 1234}
```

Построим расстояния в графе от первой вершины:

```
In[34]:= RankGraph[g, {1}]

Out[34]= {1, 2, 2, 3, 3, 2, 2, 3, 3, 4, 4, 3, 3, 4, 2, 3, 3, 4, 4, 3, 3, 2, 4, 3}
```

Рассмотрим распределение расстояний:

```
In[35]:= Distribution[RankGraph[g, {1}]]

Out[35]= {1, 6, 11, 6}
```

Отсюда следует, что шесть перестановок можно получить из тождественной перестановки одной транспозицией, одиннадцать перестановок – двумя транспозициями и шесть – тремя транспозициями.

То же самое можно вычислить с помощью числа Стирлинга первого рода:

```
In[36]:= Table[StirlingFirst[4, i], {i, 4, 1, -1}]

Out[36]= {1, 6, 11, 6}
```

В предыдущей задаче перестановки перечислялись в порядке минимального отличия друг от друга. Поставим обратную задачу:

Требуется перебрать все перестановки множества $\{1,2,\ldots,n\}$ в таком порядке, чтобы каждая последующая перестановка отличалась от предыдущей во всех позициях.

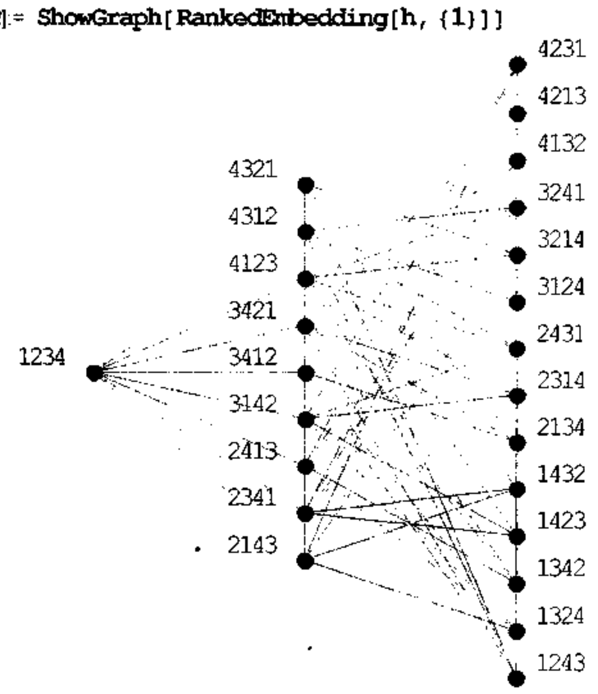
Известно, что перестановки π_1 и π_2 отличаются друг от друга во всех позициях, если *произведение* $(\pi_1)^{-1}\times\pi_2$ *есть перестановка без неподвижных точек.*

Построим булеву функцию и граф, вершинами которого являются перестановки размера 4, а две вершины смежны, если соответствующие им перестановки отличаются во всех позициях:

```
In[39]:= d = DerangementQ[Permute[InversePermutation[#1], #2]] &;
    dgraph = MakeGraph[Permutations[4], d, Type → Undirected, VertexLabel → True]

Out[40]= -Graph:<108, 24, Undirected>-
```

Чтобы метки перестановок сильно не загромождали рисунок, установим опции меток:

```
In[41]:= h = SetGraphOptions[dgraph, {1, 8, 10, 11, 14, 17, 18, 19, 23, 24, VertexLabelPosition → UpperLeft
Out[41]= -Graph:<108, 24, Undirected>-
In[42]:= ShowGraph[RankedEmbedding[h, {1}]]
```

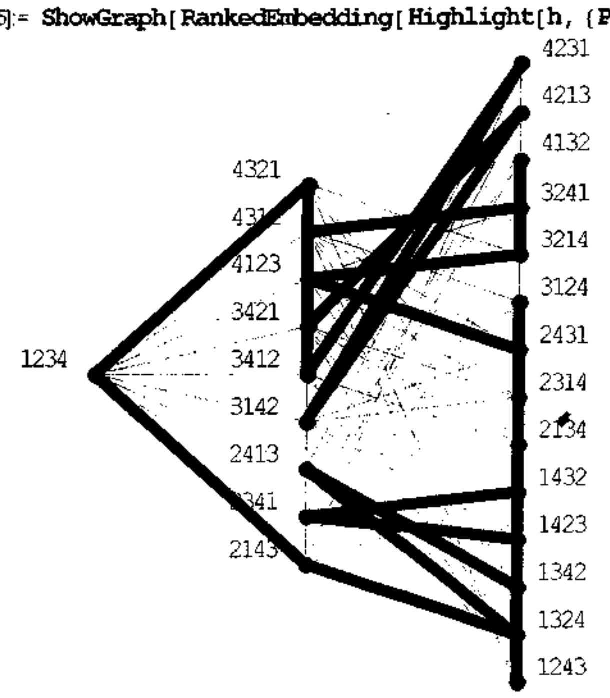


Проверим, будет ли этот граф гамильтоновым:

```
In[43]:= HamiltonianQ[h]
Out[43]= True
```

Выделим гамильтонов цикл в графе h:

```
In[44]:= l = HamiltonianCycle[dgraph]
Out[44]= {1, 8, 3, 11, 4, 7, 2, 9, 5, 10, 6, 13, 12, 19, 15, 20, 16, 23, 18, 21, 14, 22, 17, 24, 1}
In[45]:= ShowGraph[RankedEmbedding[Highlight[h, {Partition[HamiltonianCycle[h], 2, 1]}], {1}]]
```

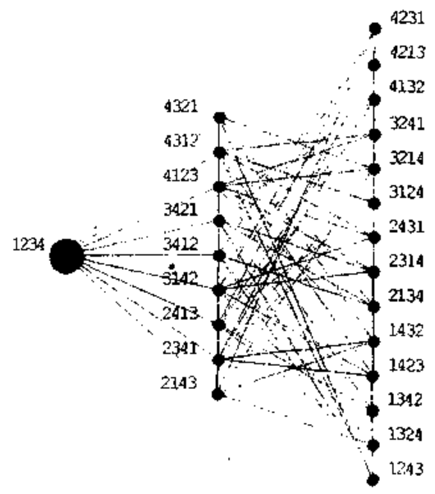


Чтобы последовательность вершин и ребер в гамильтоновом цикле была явно выделена, лучше выделить гамильтонов цикл анимацией. Для этого выделим последовательность вершин и ребер гамильтонова цикла:

```
In[46]:= s = {1, {1, 8}, 8, {8, 3}, 3, {3, 11}, 11, {11, 4}, 4, {4, 7}, 7, {7, 2}, 2, {2, 9}, 9, {9, 5}, 5, {5, 10},
    10, {10, 6}, 6, {6, 13}, 13, {13, 12}, 12, {12, 19}, 19, {19, 15}, 15, {15, 20}, 20, {20, 16}, 16,
    {16, 23}, 23, {23, 18}, 18, {18, 21}, 21, {21, 14}, 14, {14, 22}, 22, {22, 17}, 17, {17, 24}, 24, {24, 1}};
```

Эта команда в динамике покажет прохождение вершин цикла:

```
In[47]:= AnimateGraph[RankedEmbedding[h, {1}], s]
```



Дважды кликнув по этому рисунку, можно увидеть прохождение по всем вершинам и ребрам гамильтонова цикла:

```
In[48]:= GetVertexLabels[h, HamiltonianCycle[h]]

Out[48]= {1234, 2143, 1324, 2413, 1342, 2134, 1243, 2314, 1423, 2341, 1432,
    3124, 2431, 4123, 3214, 4132, 3241, 4312, 3421, 4213, 3142, 4231, 3412, 4321, 1234}
```

Эту задачу можно решить с помощью бинарного отношения Q:

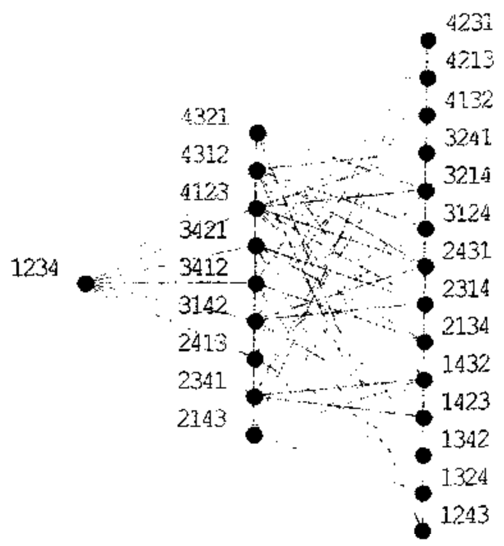
```
In[49]:= Q = (p = #1; q = #2; k = 0; Do[If[p[[i]] =!= q[[i]], , k = k + 1], {i, 1, Length[#1]}]; If[k > 0, False, True]) &;
```

Или проще:

```
In[50]:= Q = (Count[#1 - #2, 0] == 0) &;
```

Построим граф отношения Q:

```
In[51]:= ShowGraph[
    RankedEmbedding[SetGraphOptions[g = MakeGraph[Permutations[4], Q, Type → Undirected, VertexLabel → True]
      {1, 8, 10, 11, 14, 17, 18, 19, 23, 24, VertexLabelPosition → UpperLeft}], {1}], PlotRange → 0.25]
```



Выведем вершины гамильтонова цикла:

```
In[52]:= GetVertexLabels[g, HamiltonianCycle[g]]

Out[52]= {1234, 2143, 1324, 2413, 1342, 2134, 1243, 2314, 1423, 2341, 1432,
  3124, 2431, 4123, 3214, 4132, 3241, 4312, 3421, 4213, 3142, 4231, 3412, 4321, 1234}
```

Литература

1. Сурикова С.В., Анисимова М.В. Использование графовых моделей при решении задач// Нач.школа.-2000.-№4.-С. 56-62.

2. Сангалова М.Е. Формирование метода графового моделирования //Перспектива: межвуз. сб. науч. трудов.-Арзамас: АГПИ, 2003.-С.151-154.