УДК 372.851

РЕШЕНИЕ ЗАДАЧИ ПОИСКА СВЯЗНЫХ КОМПОНЕНТ ГРАФА ПОСРЕДСТВОМ СИСТЕМ СИМВОЛЬНОЙ МАТЕМАТИКИ

В.В. Васильев, В.Г. Тихомиров

ГОУ ВПО «Тамбовский государственный технический университет», г. Тамбов

Ключевые слова и фразы: дидактические единицы; комплекс учебных заданий; компонента связности; матрица смежности; орграф; теория графов; транзитивные замыкания.

Аннотация: Рассматривается ряд задач теории графов, необходимых в различных приложениях; обоснована актуальность отбора перечисленных дидактических единиц; приводится алгоритм нахождения связных компонент графа; имеется возможность контроля промежуточных результатов вычислений.

Современный рынок труда требует от высшей школы освоения все новых специальностей и направлений. Это отражается не только на профильных учебных дисциплинах, но и на общеобразовательных, в том числе и математике. Отметим, что методические аспекты, связанные с изложением традиционных для компьютерных специальностей ВУЗов разделов математики, таких, например, как математическая логика, теория алгоритмов, теория графов, нашли широкое отражение в учебной литературе. Однако, изложение, например, теории графов требует пристального методологического исследования.

Элементы теории графов в литературе часто рассматриваются абстрактно – без приведения алгоритмов решения задач. При этом обходятся стороной прикладные аспекты, в то время, как теория графов находит применение, например, при проектировании домов и сооружений, инженерных сетей, линий электропередач и т.п. Также не обходятся без применения теории графов в химии, программировании, экономике, при организации движения транспорта.

В связи с этим, нами разработан комплекс учебных заданий, на основе которых составлены компьютерные программы, генерирующие различные варианты заданий: нахождение прямого и обратного транзитивных замыканий, компонент связности; топологическая сортировка графа; поиск кратчайшего пути от вершины нижнего уровня до вершины верхнего уровня и его длины; нахождение центра графа и отклонения вершин графа от его центра; нахождение количества путей с заданным количеством дуг и самих путей; нахождение дополнительного графа, двойственного графа, реберного графа, нахождение остовов и базисных циклов; нахождение максимального паросочетания, минимального реберного покрытия, минимального вершинного покрытия, максимального независимого множества вершин.

С учетом изложенного ранее, нами выделено оптимальное количество учебных заданий, вокруг которых формируются дидактические единицы, приводящие к наибольшей эффективности усвоения содержания рассматриваемого курса. В результате формируется та часть учебной программы, где фиксируются роль и место раздела «Теория графов» в курсе математики, основные цели его изучения, содержание учебного материала, технологии организации обучения.

В качестве примера приведем задание, состоящее в следующем: даны дуги орграфа, вершины которого пронумерованы от 1 до 16, причем первое число указывает начало, второе – конец дуги. На основе аналитических выражений для прямого и обратного транзитивных замыканий необходимо найти все классы эквивалентности для графа вашего варианта. Результаты вычислений проверить путем непосредственных преобразований матриц смежности. В отчете о выполненном задании необходимо привести два рисунка: графическое изображение исходного графа, когда его вершины в порядке возрастания номеров расположены по окружности, наподобие циферблата, и этого же графа, но разложенного на сильно связанные подграфы.

Приведем основной фрагмент программы на языке «Maple». Существенно, что программы позволяют не только сгенерировать задание, но и проверить ответ и ход решения, поскольку выдают не только ответ, но и результаты промежуточных вычислений.

```
A:=RandomMatrix(n, density=0.5, generator=rand(0..1)):   G:=Graph(A):
TexMatrix(A, task):
СтроитьГраф(G, `1.bmp`);
V:={$ 1..n}: vp:=[[0, 0] $ n ]:   l:=0:
while V<>{}
do
B:={V[1]};
writeline(sols, cat(` Находим компоненту связности для вершины `, tex(op(B)), `:`)):
writeline(sols, cat(` Находим степени оператора прямого транзитивного замыкания для вершины `, tex(op(B)), `:`)):
C1:=Замыкание(A, B, n, 1, sols):
writeline(sols, cat(` Находим степени оператора обратного транзитивного замыкания для вершины `, tex(op(B)), `:`)):
C2:=Замыкание(A, B, n, -1, sols):
l:=l+1:
C3:=C1 intersect C2:
writeline(sols, cat(`Таким образом, компонента связности для вершины `, tex(op(B)), ` будет $ C_`, tex(op(B)), ` = `, tex(C3), `$.` )):
C[l]:=C3 union B:
V:=V minus C[l]:
od:
for i from 1 to l
do placevertex(C[i], i, nops(C[i]))      od:
SetVertexPositions(G, vp):
СтроитьГраф(G, `2.bmp`);  # записать изображение графа G в файл
print("vp=",vp):
```

Вначале с помощью генератора случайных чисел, распределенных равномерно, формируется матрица смежности графа. Затем с помощью процедуры «Замыкание» находятся прямое и обратное транзитивное замыкания выбранной вершины. После чего по найденным замыканиям вычисляется компонента связности текущей вершины. В соответствии с поставленным заданием процедура placevertex группирует вершины по компонентам.

Отметим, что процедура «Замыкание» не является стандартной и осуществляет поиск прямого и обратного транзитивных замыканий вершины. Кроме того, используется процедура «Строить граф», которая записывает в файл изображение графа.

Список литературы

1. Кирсанов, М.Н. Графы в Maple. Задачи, алгоритмы, программы / М.Н. Кирсанов. – М. : Издательство ФИЗМАТЛИТ, 2007.
2. Евстигнеев, В.А. Применение теории графов в программировании / В.А. Евстигнеев. – М. : Наука, 1985. – 352 с.
3. Тихомиров, В.Г. Система адаптации математической модели / В.Г. Тихомиров // Вопр. соврем. науки и практики. Ун-т им. В.И. Вернадского. – 2006. – № 2(4). – С. 77–85.