# Наглядная интерпретация некоторых алгоритмов на графах

Г. Райтер И. Сонин
(Университет Северной Каролины, г. Шарлот)

Большое место в теории графов занимают алгоритмы. Нахождение кратчайших путей, построение остовных деревьев с нужными свойствами, построение паросочетаний, раскраска вершин — вот далеко не полный перечень задач, решение которых состоит в описании алгоритма.

Некоторые студенты сталкиваются с трудностями при понимании алгоритмов. Как правило, алгоритмы на графах основаны на довольно простых идеях. Но их формальное изложение бывает длинным, а понимание такого изложения может требовать определенной математической подготовки. Еще одна трудность состоит в том, что алгоритмы зачастую записываются в псевдокоде — очень удобно для программирования компьютера, но не для восприятия человеком. Поэтому нужны какие-то мнемонические приемы для облегчения запоминания и понимания таких алгоритмов. Наш подход состоит в том, чтобы интерпретировать некоторые алгоритмы из теории графов в знакомом большинству студентов контексте: вступление в клуб.

Пусть $G(V,E)$ — граф с множеством вершин $V$ и множеством ребер $E$. Всюду далее считаем этот граф связным (в противном случае можно просто рассматривать все его связные компоненты по отдельности). Напомним, что граф $H$ называется *подграфом* $G$, если он получается выбором некоторых вершин $G$ и части соединяющих эти вершины ребер $G$. *Дерево* — это связный граф без циклов. *Остовное дерево* графа $G$ — это подграф $G$, содержащий все вершины $G$ и являющийся деревом. Мы будем рассматривать алгоритмы построения остовного дерева $T$ графа $G$. Построение остовного дерева начинается с выделенной вершины, которую будем называть *корнем* дерева.

Идея состоит в том, чтобы интерпретировать $V$ как множество людей. Ребро соединяет двух людей, если они знакомы. Есть клуб, в который хотят вступить все эти люди. Постепенно, один за другим, они туда все и вступят. Чтобы человек $u$ мог вступить в клуб, у него должен быть *поручитель*: член клуба $v$, знакомый с $u$. Член клуба может быть поручителем, если он знаком хотя бы с кем-то не из клуба, такого члена клуба будем называть *открытым*. Конечно, по мере роста клуба, открытый член клуба может перестать быть таковым, но не наоборот. Приведенные ниже правила определяют для каждого момента, кто вступает в клуб следующим и кто его поручитель.

Наше основное правило порождает широкий класс корневых остовных деревьев. Дополнительные к основному правилу условия позволяют построить такие

Перевод статьи Reiter H., Sonin I. The “Join the Club” Interpretation of Some Graph Algorithms. *College Math. Journal*, v. 27, No 1, 1996. Публикуется с любезного разрешения редакции журнала и авторов.

остовные деревья как дерево поиска в ширину (Breadth First Search, BFS), дерево поиска в глубину (Depth First Search, DFS), кратчайшая связывающая сеть (Minimal Total, MT), дерево кратчайших путей (Shortest Path, SP). Эти деревья встречаются при решении многих задач. Например, дерево поиска в ширину можно использовать для определения длины кратчайшего (содержащего наименьшее число ребер) пути из корня в любую другую вершину. Дерево поиска в глубину позволяет находить мосты в графе, а в случае их отсутствия — построить сильно связную ориентацию графа. (*Мост* — ребро, удаление которого делает граф несвязным. *Ориентация* — приписывание ребрам направления, ориентированный граф *сильно связен*, если из любой вершины можно пройти в любую по ребрам вдоль указанных на них направлений, пример см. на рис. 3.)

Основное правило. *Есть ровно один основатель клуба. Все остальные вступают в клуб по одному, пока еще есть кому вступать. Каждый вступающий в клуб имеет поручителя.*

Основное правило порождает последовательность подграфов. Начинаем с выделенного корня (основатель $u_0$), на каждом шаге в подграф добавляется одна вершина (новый член) и одно ребро (от поручителя к новому члену), рис. 1. Алгоритмы, которые мы обсудим, отличаются предписаниями по выбору нового члена и его поручителя. Если две или более пар «новый член – поручитель» имеют равный приоритет вступления в клуб, выбираем одну из них произвольно. Это соглашение применимо ко всем обсуждаемым далее алгоритмам.

Давайте убедимся, что любой подграф, получаемый в соответствии с Основным правилом и любым его уточнением, — остовное дерево. Заметим, что в любой момент множество членов клуба и ребер, соединяющих членов с их поручителями, является связным подграфом графа $G$. Предположим, что после вступления некоторого члена $u$ образовался цикл. Ясно, что $u$ не является ничьим поручителем — после него в клуб никто не вступал. Значит, оба члена клуба, связанные с $u$ ребрами цикла, — его поручители, что противоречит Основному правилу. Поэтому в любой момент мы имеем связный ациклический граф. Наконец, если в некоторый момент еще есть нечлены клуба, то — ведь граф $G$ связный — кто-то из членов клуба должен быть открытым. Так что процесс продолжается до тех пор, пока все нечлены вступят в клуб. И в этот момент клуб задает дерево, содержащее все вершины $G$, — остовное дерево $G$.

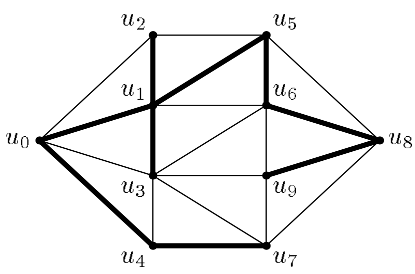


***Рис. 1.*** *Применение основного правила приводит к остовному дереву (выделено жирным). Индексы указывают на порядок вступления в клуб*

Порядок вступления в клуб задает помеченное дерево: каждой вершине приписано число от 0 до $n-1$ так, что на пути от основателя до любого члена пометки идут в возрастающем порядке.

Следующие два правила основаны на понятии *старшинства*. Если $u$ вступил в клуб раньше, чем $v$, то $u$ *старше* $v$ (а $v$ младше $u$).

Правило старшинства. *Поручителем при вступлении в клуб выбирают старейшего на данный момент открытого члена.*

Получаемое остовное дерево называется деревом поиска в ширину, рис. 2.

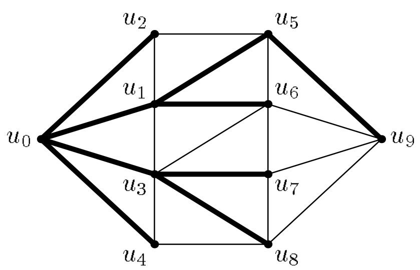


***Рис. 2.*** *Применение правила старшинства приводит к дереву поиска в ширину (выделено жирным)*

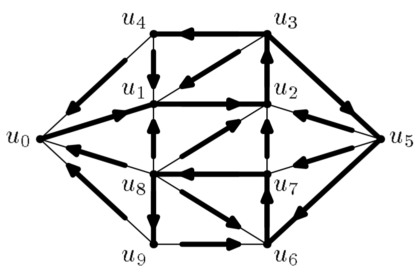


***Рис. 3.*** *Применение правила младшинства приводит к дереву поиска в глубину (выделено жирным)*

Правило младшинства. *Поручителем при вступлении в клуб выбирают самого младшего на данный момент открытого члена.*

Получаемое остовное дерево называется деревом поиска в глубину (DFS). На рис. 3 изображено DFS с дополнительной структурой сильно связной ориентации. Ребра этого дерева ориентированы от старшего члена к младшему. Остальные ребра ориентированы от младшего члена к старшему. Любой граф без мостов можно сильно связно ориентировать, построив дерево поиска в глубину и задав ориентацию как сказано выше.

Некоторые алгоритмы применяются к *сетям*, т. е. таким графам, каждому ребру $e$ которых приписан неотрицательный вес $f(e)$ (длина ребра). Кратчайший путь из вершины $u$ в вершину $v$ — это путь с наименьшей суммой весов входящих в него ребер (по сравнению с остальными путями). Чтобы сформулировать следующее правило, будем считать, что каждый член платит вступительный взнос, равный весу ребра от этого члена до его поручителя. Обратите внимание, что величина вступительного взноса может зависеть от выбора поручителя.

Первое жадное правило. *В клуб вступает тот, у кого величина вступительного взноса на данный момент является наименьшей.* (*Поручителем становится, естественно, тот, кто эту величину обеспечивает.*)

Получаемое таким образом дерево $T$ называется кратчайшей связывающей сетью (MT), см. рис. 4. Его вес — это $\sum_{e\in T} f(e)$. Жадное правило — простейший

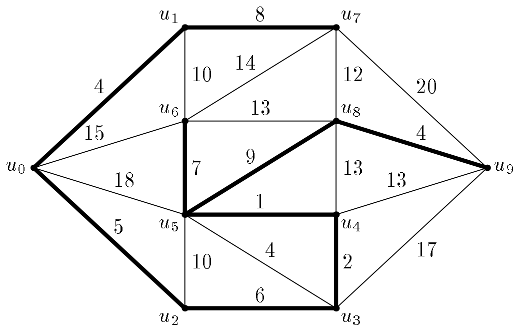


***Рис. 4.*** *Применение первого жадного правила приводит к остовному дереву наименьшего суммарного веса*

способ описать алгоритм Прима для построения остовного дерева наименьшего веса. Доказательство того, что $T$ в самом деле является МТ, есть в [3] или [5].

Наше заключительное правило описывает построение остовного дерева, задающего кратчайшие пути из корня в любую другую вершину. Чтобы его сформулировать, предположим, что каждый новый член не только должен заплатить вступительный взнос, но и вернуть своему поручителю всю сумму, которую тот заплатил при вступлении в клуб. Поэтому общая сумма, выплачиваемая новым членом, равна сумме весов ребер на пути (единственном) от основателя до этого нового члена. Как и раньше, величина выплаты зависит от выбора поручителя.

Второе жадное правило. *В клуб вступает тот, кому приходится платить меньше всего на данный момент.* (*Поручителем выбирается, естественно, тот, кто такой платеж обеспечивает.*)

Второе жадное правило — это наш способ сформулировать алгоритм Дийкстры построения остовного дерева кратчайших путей, см. рис. 5.

Все эти алгоритмы можно применять и к ориентированным графам, считая

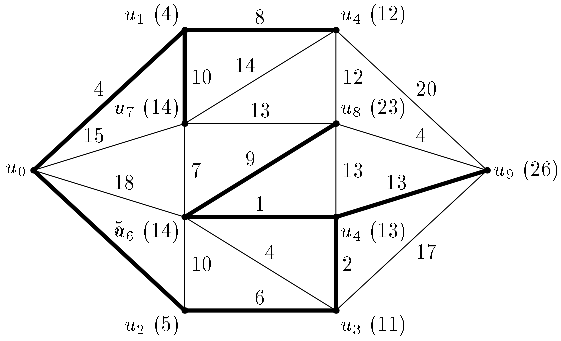


***Рис. 5.*** *Применение второго жадного правила приводит к остовному дереву с кратчайшими путями от корня до любой вершины. (В скобках указаны платежи при вступлении.)*

что ориентированное ребро $\langle u, v\rangle$ позволяет $u$ быть поручителем $v$, но не наоборот. Другие алгоритмы на графах также можно объяснять на языке клубов и членов. В частности, алгоритм Форда–Фалкерсона построения максимального потока (или минимального разреза), равно как и алгоритм построения максимального взвешенного паросочетания, использует построение остовных деревьев как часть общей процедуры, см. [3, 5]. Простой способ описания алгоритмов построения остовных деревьев с нужными свойствами облегчает понимание этих, более сложных, алгоритмов. Мы надеемся, что наши читатели с радостью вступят в клуб тех, кто знает и умеет использовать алгоритмы теории графов.

## Список литературы

[1] *K. P. Bogart.* Introductory Combinatorics. 2nd ed. Harcourt Brace Jovanovich, New York, 1990.

[2] *G. Chartand, L. Lesniak.* Graphs and Digraphs. 2nd ed. Wadsworth & Brooks/Cole, Monterey, CA, 1986.

[3] *A. Gibbons.* Algorithmic graph theory. Cambridge University Press, Cambridge, 1985.

[4] *F. Harary.* Graph theory. Addison–Wesley, Reading, MA, 1969.

[5] *J. A. McHugh* Algorithmic graph theory. Prentice Hall, Englewood Cliffs, NJ, 1990.

[6] *F. S. Roberts* Applied Combinatorics. Prentice Hall, Englewood Cliffs, NJ, 1984.

Добавлено при переводе:

[7] *А. А. Зыков.* Основы теории графов. М.: Наука, 1987.

[8] *Ф. Харари.* Теория графов. М.: Мир, 1973.

[9] *Н. Кристофидес.* Теория графов. Алгоритмический подход. М.: Мир, 1978.

[10] *А. Ахо, Дж. Хопкрофт, Дж. Ульман.* Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.