



ИТОГИ НАУКИ И ТЕХНИКИ.
Современная математика и ее приложения.
Тематические обзоры.
Том 204 (2022). С. 37–43
DOI: 10.36535/0233-6723-2022-204-37-43

УДК 519.1

КОМБИНАТОРНЫЙ АЛГОРИТМ НАХОЖДЕНИЯ КОЛИЧЕСТВА ПУТЕЙ НА ОРИЕНТИРОВАННОМ ГРАФЕ

© 2022 г. Я. М. ЕРУСАЛИМСКИЙ, М. И. ЧЕРДЫНЦЕВА

Аннотация. В работе приведен алгоритм нахождения количества путей на ориентированном графе, начинающихся в произвольном подмножестве его вершин. Алгоритм основан на идеях, лежащих в основе построения треугольника Паскаля. Трудоемкость алгоритма совпадает с трудоемкостью известного алгоритма Дейкстры нахождения кратчайших путей на графах. Также осуществлена адаптация предложенного алгоритма для решения этой задачи на графах с ограничениями на достижимость.

Ключевые слова: ориентированный граф, путь, треугольник Паскаля, ограничения на достижимость.

COMBINATORIAL ALGORITHM FOR FINDING THE NUMBER OF PATHS ON A DIRECTED GRAPH

© 2022 Ya. M. ERUSALIMSKIY, M. I. CHERDYNTSEVA

ABSTRACT. In this paper, we present an algorithm for finding the number of paths on a directed graph that start at an arbitrary subset of its vertices. The algorithm is based on the ideas underlying the construction of Pascal's triangle. The complexity of the algorithm coincides with the complexity of the well-known Dijkstra algorithm for finding shortest paths on graphs. We also generalize the algorithm proposed to the problem on graphs with reachability constraints.

Keywords and phrases: directed graph, path, Pascal's triangle, reachability constraints.

AMS Subject Classification: 05C38

1. Введение. Блез Паскаль, изучая знаменитый арифметический треугольник, носящий его имя (см., например, [5]), вероятно, не подозревал, что решает задачу о количестве путей, ведущих из фиксированной вершины графа-решетки в остальные его вершины. Граф-решетка имеет вершины в точках с целочисленными координатами в первом квадранте декартовой плоскости. Из каждой вершины графа-решетки выходит две дуги — в ближайшую правую вершину и в ближайшую верхнюю вершину (см. рис. 1).

Ясно, что для каждой его вершины, отличной от вершины $O(0; 0)$, количество путей, ведущих в нее из вершины $O(0; 0)$, равно количеству таких путей, которые заканчиваются горизонтальной дугой, плюс количество таких путей, которые заканчиваются вертикальной дугой. Именно это свойство множества путей графа-решетки порождает известное свойство треугольника Паскаля

$$\binom{n+1}{m+1} = \binom{n}{m} + \binom{n}{m+1}. \quad (1)$$

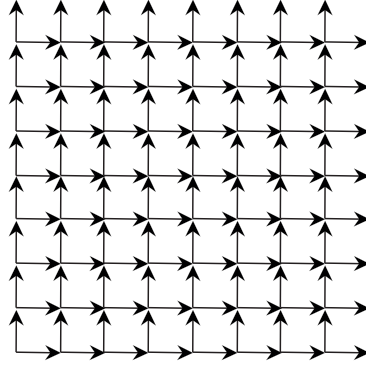


Рис. 1. Граф-решетка

Далее, отправляясь от идей Б. Паскаля, реализованных в треугольнике Паскаля, и Э. Дейкстры, реализованных в его алгоритме нахождения кратчайших путей на графе (см., например, [13]), предложим алгоритм нахождения количества путей, ведущих в вершины графа из фиксированного подмножества его вершин.

2. Обоснование и описание алгоритма. Будем пользоваться общим определением ориентированного графа (см., например, [11]), понимая под графом тройку $G(X, U, f)$, где $X (\neq \emptyset)$ — множество вершин графа, U — множество дуг графа, $f: U \rightarrow X \times X$ — отображение смежности вершин, сопоставляющее каждой дуге графа упорядоченную пару вершин. Если $f(u) = (x; y)$, то вершину x будем называть началом дуги u и писать $x = (p_1 \circ f)(u)$, а вершину y — концом дуги u и писать $y = (p_2 \circ f)(u)$.

Следуя [4], путем длины n ($n \in \mathbb{N}$) на графе $G(X, U, f)$ будем называть такое отображение $\mu: [1; n]_{\mathbb{N}} \rightarrow U$, что для любого $i \in [1; n-1]_{\mathbb{N}}$ выполнено соотношение

$$(p_2 \circ f \circ \mu)(i) = (p_1 \circ f \circ \mu)(i+1),$$

т.е. конец предыдущей дуги является началом следующей. Вершина $(p_1 \circ f \circ \mu)(1)$ называется началом пути, а вершина $(p_2 \circ f \circ \mu)(n)$ — концом пути.

Через $U_+(x)$ обозначим множество дуг графа, начинающихся в вершине x , а через $O_+(x)$ — множество вершин, в которых заканчиваются дуги из $U_+(x)$.

Через $U_-(x)$ обозначим множество дуг графа, заканчивающихся в вершине x , а через $O_-(x)$ — множество вершин, в которых начинаются дуги из $U_-(x)$.

Обозначим через $M_+(x)$ множество путей, заканчивающихся в вершине x , а через $M_{+,n}(x)$ — множество путей, имеющих длину n , заканчивающихся в вершине x .

Очевидно, что справедливы следующие утверждения.

Утверждение 1. Для любой вершины x графа $G(X, U, f)$ справедливо равенство

$$|M_+(x)| = \sum_{u \in U_-(x) | y_u = (p_2 \circ f)(u)} |M_+(y_u)|. \quad (2)$$

Здесь и далее $|A|$ — количество элементов во множестве A . Равенство (2) является аналогом соотношения Паскаля (1).

Обозначим через $M_{y,+}(x)$ множество путей, начинающихся в вершине y и заканчивающихся в вершине x , а через $M_{y,+,n}(x)$ — множество путей длины n , начинающихся в вершине y и заканчивающихся в вершине x .

Утверждение 2. Пусть $Y \subseteq X$, $x \in X$. Тогда

$$M_{Y,+}(x) = \bigcup_{y \in Y} M_{y,+}(x), \quad (3)$$

$$|M_{Y,+}(x)| = \sum_{y \in Y} |M_{y,+}(x)|, \quad (4)$$

Здесь $M_{Y,+}(x)$ — множество путей, начинающихся в вершинах множества Y и заканчивающихся в вершине x .

Утверждение 3. Пусть $Y \subseteq X$, $x \in X$, $n \in \mathbb{N}$. Тогда

$$|M_{Y,+,n}(x)| = \sum_{y \in Y} |M_{y,+,n}(x)|. \quad (5)$$

Утверждение 4. Для любой вершины x графа $G(X, U, f)$ и $n \in \mathbb{N}$ справедливо равенство

$$|M_{+,n}(x)| = \sum_{n \in \mathbb{N}} |M_{+,n}(x)|. \quad (6)$$

Замечание 1. Ясно, что утверждения 1–4 содержательны в случае, когда фигурирующие в них множества конечны. Это справедливо, в частности, для конечных графов, не содержащих контуры.

Опишем теперь динамический алгоритм, позволяющий находить для каждой вершины графа количество путей, ведущих в нее из произвольного подмножества вершин Y . Будем считать, что вершины и дуги графа занумерованы, т.е. $X = \{x_1, x_2, \dots, x_n\}$, $U = \{u_1, u_2, \dots, u_m\}$. Каждой вершине x_i поставим в соответствие пару чисел $(M_i; \nu_i)$. Эти пары образуют массив V длины n .

Для предотвращения заикливания алгоритма определим максимально допустимое количество итераций NM (причиной заикливания может быть только наличие на графе контуров).

0. Зададим начальное значение массива V :

```
for  $i \in [1, n]$  do  $M_i := 0$ ,  $\nu_i := 0$ 
for  $y \in Y$  do  $\nu_y := 1$ 
```

Найдем характеристику T_V массива V , положив

$$T_V = \sum M_i.$$

Номер итерации $N_i := 0$.

1. По массиву V построим массив V' , который устроен аналогично массиву V :

```
for  $i \in [1, n]$  do  $M'_i := M_i + \nu_i$ ,  $\nu'_i := 0$ 
for  $u_j \in U$  do  $\nu'_y := \nu'_y + \nu_x$ , где  $u_j = (x; y)$ .
```

Найдем

$$T_{V'} = \sum M'_i.$$

$N_i := N_i + 1$.

2. Пока характеристики не совпадают и не превышено допустимое количество итераций: $T_V \neq T_{V'}$ & $N_i < NM$, меняем массив $V := V'$ и повторяем построение массива V' (шаг 1).

3. Количество путей получено в массиве M_i .

Приведем еще один вариант этого алгоритма.

0. Зададим начальное значение массива V :

```
for  $i \in [1, n]$  do  $M_i := 0$ ,  $\nu_i := 0$ 
for  $y \in Y$  do  $\nu_y := 1$ 
```

Найдем характеристику R_V массива V , положив

$$R_V = \sum \nu_i.$$

Номер итерации $N_i := 0$.

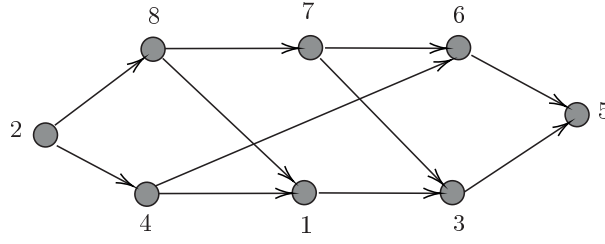


Рис. 2

Таблица 1

Номера вершин									
	1	2	3	4	5	6	7	8	T
V_0	(0; 0)	(0; 1)	(0; 0)	(0; 0)	(0; 0)	(0; 0)	(0; 1)	(0; 0)	0
V'_0	(0; 0)	(1; 0)	(0; 1)	(0; 1)	(0; 0)	(0; 1)	(1; 0)	(0; 1)	2
V_1	(0; 0)	(1; 0)	(0; 1)	(0; 1)	(0; 0)	(0; 1)	(1; 0)	(0; 1)	2
V'_1	(0; 2)	(1; 0)	(1; 0)	(1; 0)	(0; 2)	(1; 1)	(1; 1)	(1; 0)	6
V_2	(0; 2)	(1; 0)	(1; 0)	(1; 0)	(0; 2)	(1; 1)	(1; 1)	(1; 0)	6
V'_2	(2; 0)	(1; 0)	(1; 3)	(1; 0)	(2; 1)	(2; 1)	(2; 0)	(1; 0)	12
V_3	(2; 0)	(1; 0)	(1; 3)	(1; 0)	(2; 1)	(2; 1)	(2; 0)	(1; 0)	12
V'_3	(2; 0)	(1; 0)	(4; 0)	(1; 0)	(3; 4)	(3; 0)	(2; 0)	(1; 0)	17
V_4	(2; 0)	(1; 0)	(4; 0)	(1; 0)	(3; 4)	(3; 0)	(2; 0)	(1; 0)	17
V'_4	(2; 0)	(1; 0)	(4; 0)	(1; 0)	(7; 0)	(3; 0)	(2; 0)	(1; 0)	21
V_5	(2; 0)	(1; 0)	(4; 0)	(1; 0)	(7; 0)	(3; 0)	(2; 0)	(1; 0)	21
V'_5	(2; 0)	(1; 0)	(4; 0)	(1; 0)	(7; 0)	(3; 0)	(2; 0)	(1; 0)	21

1. По массиву V построим массив V' , который устроен аналогично массиву V :

for $i \in [1, n]$ do $M'_i \in M_i + \nu_i$, $\nu'_i := 0$
 for $u_j \in U$ do $\nu'_y := \nu'_y + \nu_x$, где $u_j = (x; y)$

Найдем

$$R_V = \sum \nu'_i.$$

$N_i := N_i + 1$.

2. Пока характеристика $R_{V'} \neq 0$ и не превышено допустимое количество итераций: $N_i < NM$, меняем массив $V := V'$ и повторяем построение массива V' (шаг 1).
 3. Количество путей получено в массиве M_i .

Приведем пример работы алгоритма. Рассмотрим граф, изображенный на рис. 2. Пусть $Y = \{2, 7\}$. Результаты вычислений сведем в таблицу 1.

Где же в описанном алгоритме применяются приведенные выше утверждения? В п. 1 фрагмент

for $i \in [1, n]$ do $M'_i := M_i + \nu_i$

есть применение утверждения 4, а фрагмент

for $u_j \in U$ do $\nu'_y := \nu'_y + \nu_x$, где $u_j = (x; y)$

есть применение утверждения 2.

Сделаем теперь несколько замечаний.

Замечание 2. После выполнения k -го шага алгоритма получившийся массив m_i представляет собой количества путей длины не большей, чем k , ведущих из множества Y (под длиной пути понимаем количество дуг, через которые он проходит).

Замечание 3. В силу замечания 2 алгоритм можно применять разумно (т.е. устанавливая необходимое количество итераций) и на графах, содержащих контуры.

Замечание 4. Ясно, что алгоритм применим и в случае мультиграфов, т.е. графов, имеющих кратные дуги.

3. Особенности применения алгоритма к графам с ограничениями на достижимость. Графы с ограничениями на достижимость — объекты, впервые рассмотренные в [1–3, 7, 11, 14] (обзор результатов имеется в [12]). Ограничения на достижимость означают, что на графе рассматривается не все множество путей, а некоторое его подмножество (допустимые пути). Это множество задается дополнительными требованиями на формирование последовательности дуг допустимых путей. Например, на графах со смешанной достижимостью во множестве дуг выделено непустое подмножество дуг, а последовательность дуг допустимого пути такова, что никакие две последовательные дуги пути не могут одновременно принадлежать выделенному подмножеству дуг.

В настоящее время определены и изучены различные виды ограничений на достижимость (см., например, [11]). Для этих графов решены задачи о достижимости вершин, о кратчайших путях и о случайных блужданиях по вершинам графов с ограничениями на достижимость. Характерной чертой графов с ограничениями на достижимость является неприменимость напрямую известных алгоритмов для решения классических задач о кратчайших путях, случайных блужданиях и потоковых задач. Потоковым задачам посвящены работы [6, 8–10].

Для решения перечисленных выше задач применяется метод разверток, состоящий в следующем. По исходному графу с ограничениями на достижимость строится его развертка. Множество вершин развертки представляет собой объединение множества вершин исходного графа X и нескольких его дубликатов $(X_1, X_2, \dots, X_k)$. Количество дубликатов определяется видом ограничений на достижимость. Таким образом, у каждой вершины исходного графа на развертке имеется несколько ее «экземпляров» — она сама и набор ее «двойников». Множество дуг развертки строится по множеству дуг исходного графа в соответствии с рассматриваемым видом ограничений на достижимость.

Исходная задача (о достижимости, о кратчайших путях, о случайных блужданиях, потоковая задача) переносится соответствующим образом на развертку. Развертка является графом без ограничений на достижимость и на ней возможно применение известных алгоритмов. Полученное решение на развертке «возвращается» на исходный граф с ограничениями на достижимость. С приведенным в этой работе алгоритмом можно поступить аналогично описанному, поскольку справедливо следующее утверждение.

Утверждение 5. *Количество допустимых путей, ведущих из вершины y в вершину x , на графе с ограничениями на достижимость равно количеству путей, ведущих из вершины y во множество вершин $\{x, x^1, \dots, x^k\}$, на развертке.*

Пример 1. Рассмотрим граф, изображенный на рис. 3; развертка этого графа изображена на рис. 4. Дуги, лежащие на сторонах квадрата, образуют множество дуг, по которым путь не может проходить подряд. Рассмотрим задачу о количестве допустимых путей на этом графе, ведущих из множества вершин $\{1, 3\}$. Применим на развертке алгоритм подсчета количества путей, ведущих из множества вершин $\{1, 3\}$; результаты приведены в таблице 2.

Количество смешанных путей, ведущих из множества вершин $\{1, 3\}$, на этом графе приведено во второй строке таблицы 3. В третьей строке приведено количество путей в случае отсутствия ограничения на достижимость.

В заключение сделаем еще три замечания.

1. В п. 0 присваиваем $\nu_y := 1$ для $y \in Y$, тем самым считаем, что каждая вершина соединена сама с собой путем длины ноль.

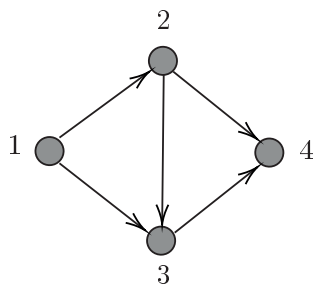


Рис. 3. Граф с ограничениями на достижимость

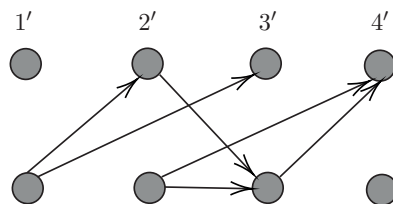


Рис. 4. Развертка графа

Таблица 2

Номера вершин									
	1	1'	2	2'	3	3'	4	4'	T_V
V_0	(0; 1)	(0; 0)	(0; 0)	(0; 0)	(0; 1)	(0; 0)	(0; 0)	(0; 0)	0
V'_0	(1; 0)	(0; 0)	(0; 0)	(0; 1)	(1; 0)	(0; 1)	(0; 0)	(0; 1)	2
V_1	(1; 0)	(0; 0)	(0; 0)	(0; 1)	(1; 0)	(0; 1)	(0; 0)	(0; 1)	2
V'_1	(1 : 0)	(0; 0)	(0; 0)	(1; 0)	(1; 1)	(1; 0)	(0; 0)	(1; 0)	5
V_2	(1 : 0)	(0; 0)	(0; 0)	(1; 0)	(1; 1)	(1; 0)	(0; 0)	(1; 0)	5
V'_2	(1; 0)	(0; 0)	(0; 0)	(1; 0)	(2; 0)	(1; 0)	(0; 0)	(1; 1)	6
V_3	(1; 0)	(0; 0)	(0; 0)	(1; 0)	(2; 0)	(1; 0)	(0; 0)	(1; 1)	6
V'_3	(1; 0)	(0; 0)	(0; 0)	(1; 0)	(2; 0)	(1; 0)	(0; 0)	(2; 0)	7
V_4	(1; 0)	(0; 0)	(0; 0)	(1; 0)	(2; 0)	(1; 0)	(0; 0)	(2; 0)	7
V'_4	(1; 0)	(0; 0)	(0; 0)	(1; 0)	(2; 0)	(1; 0)	(0; 0)	(2; 0)	7

Таблица 3

Номер вершины	1	2	3	4
Количество смешанных путей	1	1	3	2
Количество путей без ограничения на достижимость	1	1	3	4

2. Трудоемкость алгоритма не зависит от количества вершин во множестве Y вершин графа, для которых вычисляем количество путей на графе, начинающихся в этих вершинах.
3. Трудоемкость предложенного алгоритма такая же, как у алгоритма Дейкстры нахождения кратчайших путей на графе.

СПИСОК ЛИТЕРАТУРЫ

1. Басангова Е. О., Ерусалимский Я. М. Различные виды смешанной достижимости // в кн.: Алгебра и дискретная математика. — Элиста: КалмГУ, 1985. — С. 70–75.
2. Басангова Е. О., Ерусалимский Я. М. Смешанная достижимость на частично ориентированных графах // в кн.: Вычислительные системы и алгоритмы. — Ростов-на-Дону: РГУ, 1983. — С. 135–140.
3. Ерусалимский Я. М. 2–3 paths in a lattice graph. Random walks // Мат. заметки. — 2018. — 104, № 3. — С. 396–406.
4. Ерусалимский Я. М. Дискретная математика. Теория и практикум. — СПб.: Лань, 2018.

5. *Ерусалимский Я. М.* Треугольник Паскаля: комбинаторика и случайные блуждания. — М.: Вузовская книга, 2020.
6. *Ерусалимский Я. М., Петросян А. Г.* Многопродуктовые потоки в сетях с нестандартной достижимостью// Изв. вузов. Сев.-Кав. рег. Естеств. науки. Прилож. — 2005. — № 6. — С. 8–16.
7. *Ерусалимский Я. М., Скороходов В. А.* Общий подход к нестандартной достижимости на графах// Изв. вузов. Сев.-Кав. рег. Естеств. науки. — 2005. — Спецвыпуск. Псевдодифференциальные уравнения и некоторые проблемы математической физики. — С. 64–67.
8. *Ерусалимский Я. М., Скороходов В. А.* О потоках в сети с ограничениями на достижимость. Вычислительный эксперимент// Мат. Весенней Воронеж. мат. школы «Современные методы теории краевых задач» (Понтрягинские чтения–XXVI). — ВГУ, 2015. — С. 89–90.
9. *Ерусалимский Я. М., Скороходов В. А.* Потоки в сетях со связанными дугами// Изв. вузов. Сев.-Кав. рег. Естеств. науки. Прилож. — 2003. — № 8. — С. 9–12.
10. *Ерусалимский Я. М., Скороходов В. А.* Прибыль от потоков с обратной связью в орсетях с ограничениями на достижимость// Изв. вузов. Сев.-Кав. рег. Естеств. науки. Прилож. — 2003. — № 8. — С. 3–8.
11. *Ерусалимский Я. М., Скороходов В. А., Кузьмина М. В., Петросян А. Г.* Графы с нестандартной достижимостью. Задачи, приложения. — Ростов-на-Дону: ЮФУ, 2009.
12. *Жулякова Л. Ю.* Графовые динамические модели и их свойства// Автомат. телемех. — 2015. — 8. — С. 115–139.
13. *Dijkstra E. W.* A note on two problems in connexion with graphs// Numer. Math. — 1959. — 1. — P. 269–271.
14. *Erusalimskiy I. M.* Graph-lattice: random walk and combinatorial identities// Bol. Soc. Mat. Mexicana. — 2016. — 22, № 2. — P. 329–335.

Ерусалимский Яков Михайлович
Южный федеральный университет
E-mail: ymerusalimskiy@sfedu.ru

Чердынцева Марина Игоревна
Южный федеральный университета
E-mail: micherdynceva@sfedu.ru