1974 г. ОПТИМИЗАЦИЯ Выпуск 15(32)
Машинная реализация массовых вычислений

УДК 681.3.06 : 51

АЛГОРИТМ РАНЖИРОВАНИЯ ГРАФА В ПАМЯТИ С ПОСЛЕДОВАТЕЛЬНЫМ ДОСТУПОМ

И.В.Иловайский

Рассматривается задача ранжирования ориентированного графа без контуров (необязательно связного) в условиях, когда граф задан на ленте массивом упорядоченных пар смежных вершин (массивом дуг) в произвольном порядке, причем размер оперативной памяти пренебрежимо мал по сравнению с длиной массива, а лента организована как память с последовательным доступом.

В памяти с произвольным доступом задача ранжирования сводится,по сути,к поиску необходимых компонент, при этом исходный массив может быть записан в произвольном порядке. В памяти с последовательным доступом возможен лишь один вид поиска - - последовательный просмотр, поэтому организация просматриваемого массива играет решающую роль. Задача, таким образом, сводится к нахождению требуемого вида массива. Как только вид массива установлен, его несложно получить с помощью процедур упорядочения и поэлементного сравнения.

Итак, исходный массив M_0 есть последовательность пар $\langle(i,j),\ldots,(l,k)\rangle$, где i, j, ..., l, k - имена вершин графа (произвольные натуральные числа). Окончательный массив должен иметь вид: $\langle((p,r),\ldots,(m,n))_1,\ldots,((x,y),\ldots,(u,v))_s\rangle$, где индексированные скобки содержат пары, все левые компоненты которых входят в уровень, имеющий тот же номер, что и индекс при скобке.

Мы не можем решить задачу построения такого массива одними упорядочениями, ибо в именах вершин отсутствуют указатели уров-

ней. По сути дела, граф задан в виде цепного списка, но вместо адреса отсылки мы должны искать следующий элемент по имени. Строить поуровневое описание мы можем лишь последовательно, идя, например, от входных (в случае дерева-корневых) вершин к выходным (висячим). Для этого построения необходим признак, по которому на каждом шаге могут выделяться вершины, входящие в очередной уровень.

Для первого шага этот признак следующий. Номера вершин I-го уровня (для деревьев - корневые вершины) встречаются в исходном массиве только как левые компоненты пар. Изъяв такие пары, мы получим массив, у которого тем же признаком обладают вершины 2-го уровня, и т.д. Недостаток этого признака - неудобство применения. Мы ведь не знаем, какие имена (числа) суть имена вершин I-го уровня, и будем вынуждены перебрать все. Чтобы избежать этого, введем в рассмотрение массив $\bar{M}_o$ обратных пар, в котором каждой паре из M_o соответствует пара с переставленными компонентами. Очевидно, если рассматривать левые компоненты пар обоих массивов, то те имена, которые встретятся в обоих массивах, принадлежат промежуточным вершинам, те, что только в M_o - вершинам I-го уровня, и те, что только в $\bar{M}_o$ - висячим (концевым) вершинам. Для исключения промежуточных вершин и обнаружения начальных (корневых) удобно массивы M_o и $\bar{M}_o$ упорядочить лексикографически по левым компонентам пар.

Процесс нахождения вершин I-го уровня (или i -го после изъятия вершин уровней с I-го по (i - I)-й) легко построить, если рассматривать M_o и соответственно $\bar{M}_o$ как массивы имен левых компонент, т.е. как массивы целых чисел. Тогда пусть M_o' и $\bar{M}_o'$ - упорядоченные по возрастанию последовательности этих целых чисел, быть может, с повторениями. Если какое-либо число (имя) t встречается в M_o' и не встречается в $\bar{M}_o'$, оно относится к I-му (или после изъятия имён вершин с I-го по (i-I)-й, к i-му) уровню. Если оно встречается в M_o' и $\bar{M}_o'$, то это имя промежуточной вершины. Если встречается в $\bar{M}'$, но не в M_o' -концевой (для дерева - висячей).

Пусть в процессе просмотра мы дошли до $n_i \in M_o'$ и $m_j \in \bar{M}_o'$ и пусть $n_i = m_j$. По условию, мы не можем обозревать более одного элемента последовательности и не можем возвращаться назад, поэтому запомним n_i . Рассмотрим n_{i+1} и m_{j+1} . Если они равны, сравним их с n_i . Если $n_{i+1} = m_{j+1} = n_i$, мы находим-

ся в ситуации, с которой начали рассуждение. Если $n_{i+1} = m_{j+1} \neq n_i$, запомним n_{i+1} и перейдем к паре n_{i+2}, m_{j+2}. Если же $n_{i+1} \neq m_{j+1}$, то возможны две ситуации. Во-первых, $n_{i+1} < m_{j+1}$. Если в этом случае $n_{i+1} = n_i$, рассмотрим пару n_{i+2}, m_{j+1}, предварительно запомнив n_{i+1}. В противном случае n_{i+1} отнесем к множеству N' имен выделяемого (I-го или i-Го) уровня и обратимся к паре n_{i+2}, m_{j+2}.

Если $n_{i+1} > m_{j+1}$, будем поступать аналогично, но в последнем случае m_{j+1} относится не к выделяемому уровню, а к множеству концевых вершин. Поэтому просто перейдем к паре n_{i+1}, m_{j+2}.

После изъятия из M_0 пар, чьи левые компоненты принадлежат очередному выделяемому уровню, соответствующие пары (инверсии) надо изъять из $\bar{M}_0$. Массив инверсных пар $\bar{N}$ упорядочивается и сравнивается с $\bar{M}_0$. Все встреченные одинаковые пары изымаются. Ввиду одинаковости порядка в массивах, процедура отыскания совпадающих пар очевидна.

О ц е н к а . Число циклов выделения вершин всех уровней равно числу уровней u. В каждом цикле массивы M_0 и $\bar{M}_0$ просматриваются один раз для формирования массива вершин выделяемого уровня N. Массив N перестраивается в массив инверсий $\bar{N}$, который просматривается совместно с $\bar{M}_0$ на предмет изъятия из последнего пар с именами вершин выделяемого уровня. Заметим, что с каждым циклом длина M_0 и $\bar{M}_0$ убывает. Общее число прогонов массивов, таким образом, равно $4u$. Эта оценка не включает число прогонов при сортировке массива $\bar{N}$ и при предварительном упорядочении M_0 и $\bar{M}_0$.

Если в графе есть контуры, то наступит момент, когда в массиве M_0 не будет пар, чьи левые компоненты не встречались бы в $\bar{M}_0$. Таким образом, алгоритм обладает способностью обнаруживать ошибки в структуре исходных данных.

ПРИМЕР. На рис. I приведен ранжированный граф. Пусть исходное описание графа массивом дуг имеет вид:

$$M_0 = (ha, ve, hp, si, uk, ul, eb, km, mc, ca, kd, dn,$$
$$zv, zw, yx, tu, ts, wf, fh, id, qb, rq, qr, xq, xf).$$

Тогда массив обратных пар выглядит следующим образом:

$$\bar{M}_0 = (ah, ev, ph, is, ku, lu, be, mk, cm, ac, dk, nd,$$

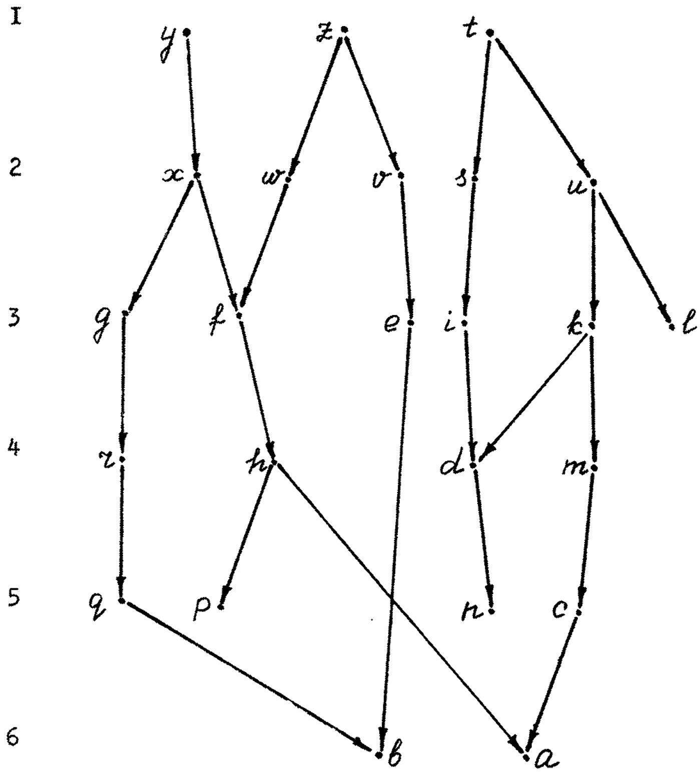

Рис.I.

$$vz, wz, xy, ut, st, fw, hf, di, bq, qr, rq, gx, fx).$$

После лексикографического упорядочения M_0 и $\bar{M}_0$ принимают вид:

$$M_0 = (\underset{V}{ca}, \underset{IV}{dn}, \underline{\underline{\underline{eb}}}, \underline{\underline{\underline{fh}}}, \underline{\underline{\underline{qr}}}, \underset{IV}{ha}, \underset{IV}{hp}, \underline{\underline{\underline{id}}}, \underline{\underline{\underline{km}}}, \underline{\underline{\underline{kd}}}, \underset{IV}{mc}, \underset{V}{qb}, \underset{IV}{rq},$$

$$\underline{\underline{si}}, \underline{tu}, \underline{ts}, \underline{\underline{uk}}, \underline{\underline{ul}}, \underline{\underline{ve}}, \underline{\underline{wf}}, \underline{\underline{xg}}, \underline{\underline{xf}}, \underline{yx}, \underline{zv}, \underline{zw});$$

$$\bar{M}_0 = (\underset{IV}{ah}, \underset{V}{ac}, \underline{\underline{\underline{be}}}, \underset{V}{bq}, \underset{IV}{cm}, \underline{\underline{\underline{dk}}}, \underline{\underline{\underline{di}}}, \underline{\underline{ev}}, \underline{\underline{fw}}, \underline{\underline{fx}}, \underline{\underline{gx}}, \underline{\underline{\underline{hf}}}, \underline{\underline{is}},$$

$$\underline{\underline{ku}}, \underline{\underline{lu}}, \underline{\underline{\underline{mk}}}, \underset{IV}{nd}, \underset{IV}{ph}, \underset{IV}{qr}, \underline{\underline{\underline{rg}}}, \underline{st}, \underline{ut}, \underline{vz}, \underline{wz}, \underline{xy}).$$

После I-го шага

$$N_1 = (tu, ts, yx, zv, zw); \quad \bar{N}_1 = (ut, st, xy, vz, wz).$$

M_0 и $\bar{M}_0$ не будут иметь пар с одним подчеркиванием.

После 2-го шага

$$N_2 = (si, uk, ul, ve, wf, xf, xd); \quad \bar{N}_2 = (is, ku, lu, ev, fw, fx, gx).$$

M_0 и $\bar{M}_0$ не будут иметь пар с двойным подчеркиванием.

После 3-го шага

$$N_3 = (eb, fh, qr, id, km, kd); \quad \bar{N}_3 = (be, hf, rg, di, mk, dk).$$

После 4-го шага

$$N_4 = (dn, ha, hp, mc, rq); \quad \bar{N}_4 = (nd, ah, ph, cm, qr).$$

После 5-го шага

$$N_5 = (ca, qb); \quad \bar{N}_5 = (ac, bq).$$

Оба массива, M_0 и $\bar{M}_0$, исчерпаны.

Сличение с рис. I показывает правильность сортировки по уровням.

Заметим, что процедура получения массива i-го уровня аналогична процедуре, применяемой при сортировке слиянием, если считать, что ключ имеет два значения: "принадлежать выделяемому уровню" и "не принадлежать". Как упоминалось, упорядочение сразу для всех уровней провести невозможно из-за отсутствия в исходном массиве соответствующих признаков в явном виде.

Принято считать, что основу для упорядочения записи образует к л ю ч с о р т и р о в к и – поле в записи, значения которого используются для фиксации относительного адреса записи [I]. Описанная задача относится к классу задач упорядочения, однако в исходной записи ключ отсутствует, и основной нашей целью было формирование массива в такой форме, чтобы этот ключ появился, т.е. приписывание ключей сортировки к записям (объектам). Оказалось, что в случае цепного представления данных (списки дуг графа) в памяти с последовательным доступом процесс приписывания ключей совмещается с последующим упорядочением по рангам, и последнее упрощается. Об этом говорит и оценка, существенно лучшая, чем для сортировки слиянием. Однако процесс требует предварительного упорядочения по п р о и з в о л ь н о м у к л ю ч у.

Л и т е р а т у р а

I. МИДОУ Ч. Анализ информационно-поисковых систем. "Мир", М., 1970.

Поступила в ред.-изд. отд.
II. XI. 1974 г.