G. M. Adel'son-Vel'skii, E. M. Landis, An algorithm for organization of information, *Dokl. Akad. Nauk SSSR*, 1962, Volume 146, Number 2, 263–266

Доклады Академии наук СССР
1962. Том 146, № 2

МАТЕМАТИКА

Г. М. АДЕЛЬСОН-ВЕЛЬСКИЙ, Е. М. ЛАНДИС

ОДИН АЛГОРИТМ ОРГАНИЗАЦИИ ИНФОРМАЦИИ

(Представлено академиком И. Г. Петровским 17 IV 1962)

В заметке будет идти речь об организации информации, расположенной в ячейках автоматической вычислительной машины. Для определенности будет рассматриваться трехадресная машина.

Постановка задачи. В машину последовательно поступает информация из некоторого запаса. Элемент информации содержится в группе ячеек, расположенных подряд. В элементе информации содержится некоторое число — оценка информации,— различное для различных элементов. Требуется организовать размещение информации в памяти машины так, чтобы в любой момент поиск информации с данной оценкой и занесение нового элемента информации требовали не очень большого числа действий.

В заметке предлагается алгоритм, где как поиск, так и занесение производятся за $C \lg N$ действий, где N — число элементов информации, поступивших к данному моменту.

Для хранения поступающей информации отводится часть памяти машины. Элементы информации кладутся туда в порядке поступления. Кроме того, в другой части памяти создается «справочный стол» ([1]), каждая ячейка которого соответствует одному из элементов информации.

Справочный стол есть диадическое дерево (рис. 1*а*): у каждой его ячейки есть не более чем одна непосредственно подчиненная ей левая и не более чем одна непосредственно подчиненная ей правая ячейка. Непосредственная подчиненность индуцирует подчиненность (частичную упорядоченность). При этом для каждой ячейки дерева все ячейки, подчиненные левой (правой) непосредственно подчиненной, будут расположены левее (правее) данной ячейки. Кроме того, мы считаем, что существует ячейка, которой подчинены все остальные (голова). По транзитивности понятие «левее» и «правее» распространяется на множество всех пар ячеек, и это множество становится упорядоченным. Так, определенный порядок ячеек в справочном столе должен совпадать с порядком расположения оценок соответствующих элементов информации (для определенности будем считать оценки возрастающими слева направо).

В первом адресе каждой ячейки справочного стола указано место, где расположен соответствующий элемент информации. Во втором и третьем адресах расположены адреса ячеек справочного стола, непосредственно подчиненных данной ячейке соответственно слева и справа. Если у ячейки с какой-нибудь стороны нет непосредственно подчиненных, то в соответствующем адресе — нуль. В некоторой фиксированной ячейке l хранится адрес головы.

Назовем цепочкой последовательность ячеек дерева, в которой каждая последующая непосредственно подчинена предыдущей. Для каждой ячейки дерева назовем длиной левой (правой) ветви максимальную длину цепочки, состоящей из ячеек, подчиненных данной и расположенных левее (правее) данной ячейки. Любая цепочка, длина которой равна длине ветви, называется стержнем ветви.

Допустимым деревом назовем такое, что для каждой его ячейки длина левой ветви отличается от длины правой не более чем на единицу (рис. 1*б*).

В каждой ячейке адресного стола в кодовой части выделяются два разряда для информации о длинах ветвей. Если левая ветвь длиннее правой, то в этих разрядах стоит 1,0. Если правая ветвь длиннее левой, то 0,1, и если они равны, то 0,0. При этом считается, что если с какой-нибудь стороны нет подчиненных ячеек, то длина ветви с этой стороны равна нулю.

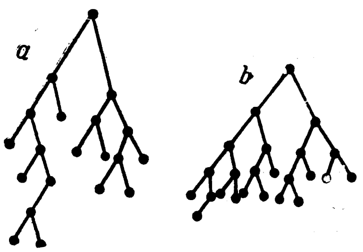

Рис. 1

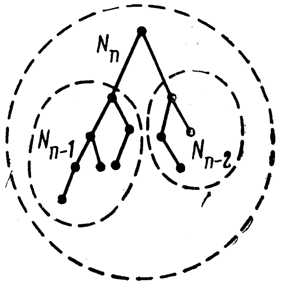

Рис. 2

Алгоритм занесения таков, что в каждый момент справочный стол является допустимым деревом.

Лемма 1. *Пусть число ячеек допустимого дерева равно N. Тогда максимальная длина ветви не больше чем* $^3/_2 \log_2 (N+1)$.

Доказательство. Обозначим через N_n минимальное число ячеек в допустимом дереве при данной максимальной длине ветви n. Тогда легко доказывается (см. рис. 2), что $N_n = N_{n-1} + N_{n-2} + 1$.

Решая это уравнение в конечных разностях, получаем

$$N_n = \left(1+\frac{2}{\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n + \left(1-\frac{2}{\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^n - 1.$$

Отсюда

$$n < \log_{\frac{1+\sqrt{5}}{2}} (N+1) < \frac{3}{2}\log_2 (N+1),$$

что и требовалось доказать.

Алгоритм поиска элемента информации с данной оценкой заключается в следующем. Сравниваем заданную оценку с оценкой элемента информации, соответствующего голове. В зависимости от результата сравнения мы переходим к сравнению заданной оценки с оценкой элемента информации, соответствующего левой или правой непосредственно подчиненной голове ячейки. Пусть сделано k шагов сравнения и на k-м шагу производилось сравнение заданной оценки m с оценкой m_u элемента информации, соответствующего некоторой ячейке u справочного стола. Если $m < m_u$ ($m > m_u$), то на $(k+1)$-м шагу производится сравнение оценки m с оценкой элемента информации, соответствующего ячейке, непосредственно подчиненной u слева (справа). Если $m = m_u$, поиск закончен.

Из единственности головы следует, что если среди накопленной информации имеется элемент информации, оценка которого равна заданной, то на некотором шагу он будет найден. Число сравнений при этом будет равно числу ячеек некоторой цепочки дерева (считая, что сравнение дает 3 ответа). Так как число действий пропорционально числу таких сравнений, то из леммы 1 следует, что число действий, потребных в этом алгоритме для поиска, не превосходит $C \log_2 (N+1)$.

Перейдем к описанию алгоритма построения справочного стола в виде допустимого дерева.

Дерево строится по мере поступления информации следующим образом. При поступлении первого элемента информации в некоторой ячейке в первом адресе указывается место расположения этого элемента в памяти. В остальные адреса и в выделенные разряды кода заносятся нули. Эта ячейка объявляется головой. Соответственно ее адрес заносится в выделенную для этого ячейку. Пусть построено допустимое дерево для N элементов информации и поступил $(N+1)$-й элемент информации. Применяем алгоритм поиска к оценке m этого элемента, запоминая при этом адреса ячеек цепочки, по которой мы проходим при этом алгоритме. Эту цепочку в дальнейшем мы будем называть о т м е ч е н н о й. Если окажется, что новый элемент уже содержится среди прежней информации, то дерево не изменяется. В противном случае мы дойдем до ячейки u, обладающей следующим свойством. Если $m < m_u$ $(m > m_u)$, где m_u — оценка элемента информации, соответствующего u, то у ячейки u нет непосредственно подчиненного слева (справа). Тогда в справочный стол добавляется новая ячейка v, непосредственно подчиненная слева (справа) ячейке u. В первом адресе ячейки v указывается место, где располагается новый элемент информации; в остальных адресах и в выделенных разрядах стоят нули. В соответствующем адресе ячейки u помещается адрес ячейки v.

Таким образом, справочный стол, пополненный ячейкой v, остается деревом, но, может быть, недопустимым. Кроме того, выделенные разряды в ячейках отмеченной цепочки нуждаются в корректировке.

Прежде всего производится корректировка выделенных разрядов в ячейке u. Имеются две возможности: 1) в выделенных разрядах ячейки u стояло 1 0 (0 1) и 2) в этих разрядах стояло 0 0. В первом случае к ячейке u добавляется новая ветвь со свободной стороны, при этом длины обеих ветвей становятся одинаковыми (равными 1), а длина любой другой ветви в дереве не изменяется. Поставив в выделенных разрядах ячейки u 0 0, мы завершаем занесение, так как полученное дерево допустимо. Во втором случае мы меняем выделенные разряды u на 1 0, если $m < m_u$, и на 0 1, если $m > m_u$.

Л е м м а 2. *Цепочка $\mathfrak{C}$ допустимого дерева, во всех ячейках которой в выделенных разрядах стоит* 0 0, *а последняя ячейка не имеет подчиненных* (*во* 2-*м и* 3-*м адресах стоят нули*), *является стержнем ветви для ячейки, непосредственно предшествующей первой ячейке w цепочки $\mathfrak{C}$.*

Д о к а з а т е л ь с т в о. Допустим, что лемма неверна. Из совокупности всех цепочек, начинающихся от w и имеющих длину бо́льшую, чем длина $\mathfrak{C}$, выберем цепочку $\mathfrak{D}$, имеющую с $\mathfrak{C}$ максимальное число общих ячеек. Пусть t — последняя общая ячейка цепочек $\mathfrak{C}$ и $\mathfrak{D}$. Тогда ее ветви имеют разную длину; следовательно, в выделенных разрядах ячейки t не может стоять 0 0. Таким образом, лемма 2 доказана.

Рассмотрим теперь максимальную связную часть $\mathfrak{C}$ отмеченной цепочки, идущую вверх, начиная с ячейки u, состоящую из ячеек, в выделенных разрядах которых стоит 0 0. Корректировка выделенных разрядов в ячейках $\mathfrak{C}$ состоит в том, что 0 0 изменяется на 1 0 или 0 1 в зависимости от того, является ли следующая ячейка цепочки $\mathfrak{C}$ непосредственно подчиненной слева или справа.

Из леммы 2 следует, что: 1) ячейки, подчиненные любой ячейке из $\mathfrak{C}$, образуют допустимое дерево; 2) для ячеек из $\mathfrak{C}$ корректировка выделенных разрядов произведена правильно и 3) длина любой ветви, содержащей ячейки из отмеченной цепочки, увеличилась на 1.

Заметим, кроме того, что ячейки, подчиненные любой ячейке, не принадлежащей отмеченной цепочке, образуют допустимое дерево, не изменившееся от добавления ячейки v.

Возможны 3 случая:

1) $\mathfrak{C}$ является всей отмеченной цепочкой. Тогда мы уже имеем допустимое дерево.

2) $\mathfrak{C}$ лежит в короткой ветви непосредственно ей предшествующей ячейки s_0. Тогда после добавления ячейки v получилось допустимое дерево, и в выделенных разрядах ячейки s_0 надо лишь поставить 0 0.

3) $\mathfrak{C}$ лежит в длинной ветви ячейки s_0. После прибавления ячейки v эта ветвь ячейки s_0 стала на 2 длиннее короткой ветви. Рассмотрим три возможных (с точностью до симметрии) расположения ячеек, подчиненных s_0, изображенные на рис. 3. A, B, C, D, E, F — ветви, не изображенные на чертеже. Число в скобках означает длину соответствующей ветви.

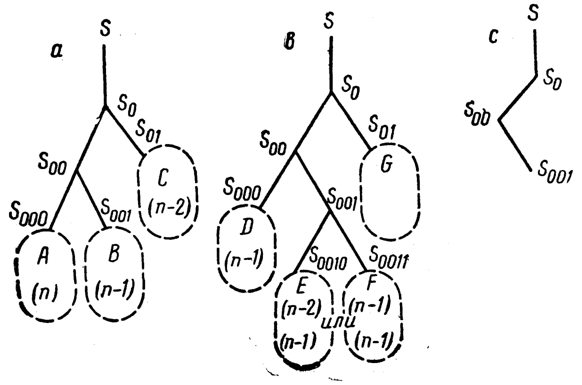

Рис. 3

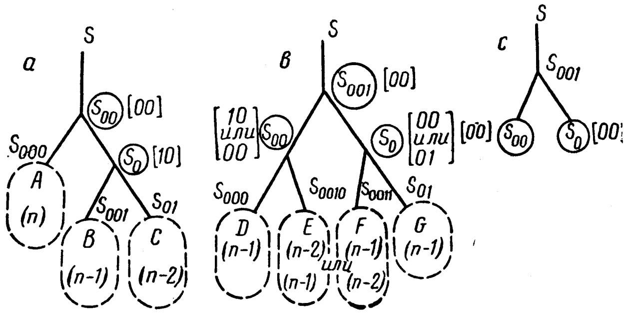

Рис. 4

В этом случае производится перестройка дерева, указанная на рис. 4. s — ячейка, непосредственно предшествующая s_0. Кружками обведены ячейки, у которых изменяются указания о непосредственно подчиненных ячейках. В квадратных скобках указаны значения выделенных разрядов в соответствующих ячейках, если их нужно изменить. После этой перестройки, как это видно из чертежа, дерево становится допустимым. Если s_0— голова (ячейки s при этом нет), то нужно еще изменить фиксированную ячейку, в которой указан адрес головы.

Так как поиск требует $C \log_2 (N + 1)$ действий, движение вверх по отмеченной цепочке вплоть до ячейки s_0 требует не более $C_1 \log_2 (N + 1)$ действий и, наконец, перестройка дерева требует постоянного числа действий, то всего требуется $C \log_2 (N + 1)$ действий.

Поступило
13 IV 1962

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

[1] Windley, Comp. J., 3, № 2, 84 (1960).