Нетрудно видеть, что описанное выше построение обобщается для любого $t \geq 2$. В полном t-арном дереве с внутренними узлами $\{1, 2, \ldots, n\}$ родителем узла k будет узел

$$\lfloor (k+t-2)/t \rfloor = \lceil (k-1)/t \rceil,$$

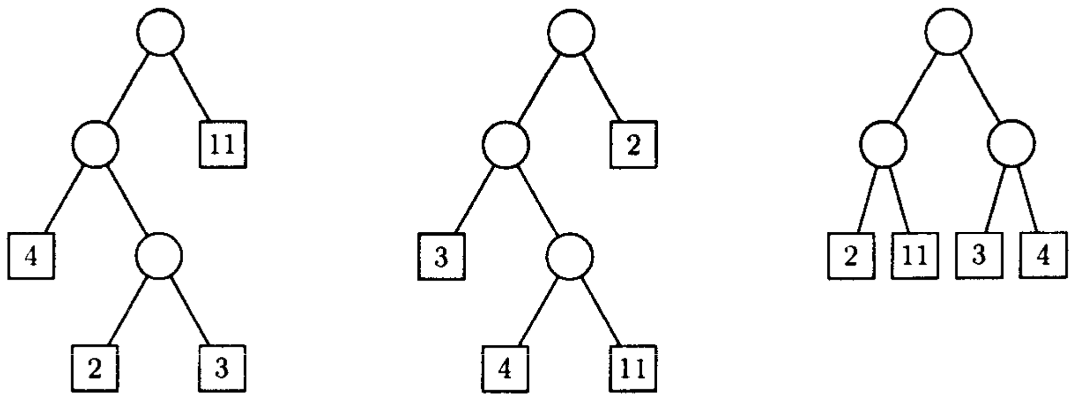
а детьми узла k — узлы

$$t(k-1)+2, \quad t(k-1)+3, \quad \ldots, \quad tk+1.$$

Это дерево имеет минимальную внутреннюю длину пути среди всех t-арных деревьев с n внутренними узлами. В упр. 8 приводится доказательство того, что его внутренняя длина пути равна

$$\left(n + \frac{1}{t-1}\right)q - \frac{(t^{q+1}-t)}{(t-1)^2}, \qquad q = \lfloor \log_t((t-1)n+1) \rfloor. \tag{7}$$

Эти результаты имеют еще одно важное обобщение, если рассмотреть их с несколько другой точки зрения. Пусть даны m действительных чисел w_1, w_2, $\ldots$, w_m. Задача заключается в поиске расширенного бинарного дерева с m внешними узлами и такого соответствия между числами $w_1, \ldots, w_m$ и узлами этого дерева, при котором сумма $\sum w_j l_j$ является минимальной, где l_j — длина пути от корня, а сумма берется по всем внешним узлам. Например, если заданы числа 2, 3, 4, 11, можно построить три таких расширенных бинарных дерева.



(8)

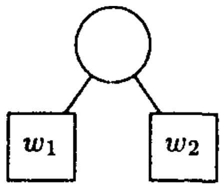
Здесь "взвешенными" длинами пути $\sum w_j l_j$ будут 34, 53 и 40 соответственно. (Как видно из данного примера, с помощью полностью сбалансированного дерева *нельзя* получить минимальное значение взвешенной длины пути для весов 2, 3, 4 и 11, хотя, как показано выше, это возможно в особом случае, с весами $w_1 = w_2 = \cdots = w_m = 1$.)

В разных компьютерных алгоритмах понятие взвешенной длины пути может интерпретироваться по-разному. Например, с его помощью можно выполнять слияние упорядоченных последовательностей с длинами $w_1, w_2, \ldots, w_m$ (см. главу 5). Одно из наиболее непосредственных приложений этого понятия заключается в том, что бинарное дерево рассматривается как некая программа поиска. Поиск начинается в корне с проверкой некоторого условия, затем в зависимости от его результата происходит переход к одной из двух ветвей, где снова проверяется некоторое условие, и т. д. Например, если необходимо выполнить проверку истинности четырех различных условий, а вероятности их истинности равны соответственно $\frac{2}{20}$, $\frac{3}{20}$, $\frac{4}{20}$ и $\frac{11}{20}$, то дерево, которое минимизирует взвешенную длину пути, и будет представлять собой *оптимальную программу поиска* (*optimal search procedure*). [Эти вероятности равны указанным в (8) весам, если умножить их на нормировочный множитель 20.]

Следующий элегантный алгоритм поиска дерева с минимальной взвешенной длиной пути был предложен Д. Хаффмэном [D. Huffman, *Proc. IRE* **40** (1952), 1098–1101]. Сначала нужно найти два наименьших веса w, например w_1 и w_2. После этого задача решается для $m-1$ весов w_1+w_2, w_3, …, w_m, причем в ее решении узел

$$\boxed{w_1+w_2} \tag{9}$$

заменяется узлом

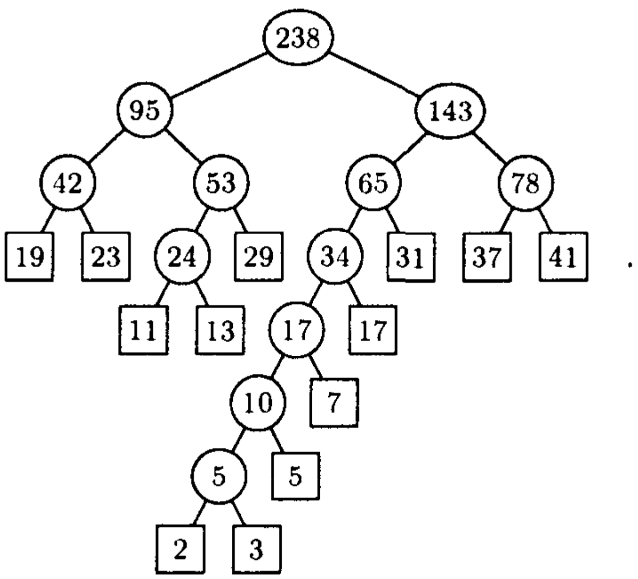


(10)

В качестве примера использования метода Хаффмэна найдем оптимальное дерево для весов 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41. Для этого сначала объединим вершины $2+3$ и найдем решение для $5, 5, 7, \ldots, 41$, затем объединим $5+5$ и т. д. В общем, последовательность действий будет выглядеть так.

$\underline{2}$	$\underline{3}$	5	7	11	13	17	19	23	29	31	37	41
	$\underline{5}$	$\underline{5}$	7	11	13	17	19	23	29	31	37	41
		$\underline{10}$	$\underline{7}$	11	13	17	19	23	29	31	37	41
			17	$\underline{11}$	$\underline{13}$	17	19	23	29	31	37	41
			$\underline{17}$		24	$\underline{17}$	19	23	29	31	37	41
					24	34	$\underline{19}$	$\underline{23}$	29	31	37	41
					$\underline{24}$	34		42	$\underline{29}$	31	37	41
						$\underline{34}$		42	53	$\underline{31}$	37	41
								42	53	65	$\underline{37}$	$\underline{41}$
								$\underline{42}$	$\underline{53}$	65		78
									95	$\underline{65}$		$\underline{78}$
									$\underline{95}$			$\underline{143}$
												238

Следовательно, такому построению Хаффмэна будет соответствовать дерево



(11)

(Числа в круглых узлах показывают связь между деревом и этапами приведенного выше вычисления; см. также упр. 9.)

Нетрудно доказать с помощью метода индукции по m, что этот способ действительно позволяет минимизировать взвешенную длину пути. Допустим, что даны веса $w_1 \le w_2 \le w_3 \le \cdots \le w_m$, где $m \ge 2$, и дерево, которое минимизирует взвешенную длину пути. (Такое дерево должно существовать, так как существует только конечное множество бинарных деревьев с m концевыми узлами.) Пусть V — внутренний узел, который находится на максимальном расстоянии от корня. Если веса w_1 и w_2 еще не приписаны детям узла V, то ими можно заменить величины, которые уже там находятся, не увеличивая взвешенную длину пути. Таким образом, существует дерево, которое минимизирует взвешенную длину пути и содержит поддерево (10). Теперь можно легко доказать, что взвешенная длина пути подобного дерева будет минимальной тогда и только тогда, когда это дерево с поддеревом (10), замененным узлом (9), обладает минимальной длиной пути для весов $w_1 + w_2$, $w_3, \ldots, w_m$. (см. упр. 9).

Всякий раз, когда в построении объединяются два веса, они по крайней мере не меньше весов, которые объединялись на предыдущем этапе, если все w_i — неотрицательные числа. Это значит, что существует прекрасный способ поиска дерева Хаффмэна при условии, что веса расположены в порядке неубывания. Тогда достаточно создать две очереди, одна из которых будет содержать исходные веса, а другая — объединенные веса. На каждом этапе этой процедуры наименьший неиспользованный вес будет находиться в начале одной из очередей, поэтому его не придется искать. В упр. 13 показано, как реализовать эту процедуру при работе с отрицательными весами.

Вообще, существует много деревьев, которые минимизируют $\sum w_j l_j$. Если в описанном выше алгоритме при очередном объединении весов всегда используется исходный, а не комбинированный вес, то полученное с помощью этого алгоритма дерево будет иметь наименьшее значение величин $\max l_j$ и $\sum l_j$ среди всех деревьев, которые минимизируют $\sum w_j l_j$. Если веса принимают положительные значения, это дерево также минимизирует $\sum w_j f(l_j)$ для *любой* выпуклой функции f по всем таким деревьям. [См. E. S. Schwartz, *Information and Control* **7** (1964), 37–44; G. Markowsky, *Acta Informatica* **16** (1981), 363–370.]

Метод Хаффмэна можно обобщить для t-арных, а также для бинарных деревьев (см. упр. 10). Еще одно важное обобщение метода Хаффмэна рассматривается в разделе 6.2.2. Обсуждение длины пути будет продолжено в разделах 5.3.1, 5.4.9 и 6.3.

УПРАЖНЕНИЯ

1. [*12*] Существуют ли какие-либо другие бинарные деревья с 12 внутренними узлами и минимальной длиной пути, кроме полного бинарного дерева (5)?

2. [*17*] Нарисуйте схему расширенного бинарного дерева с концевыми узлами, которые содержат веса 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, имеющие минимальную взвешенную длину пути.

▶ **3.** [*M24*] Расширенное бинарное дерево с m внешними узлами определяет множество длин пути $l_1, l_2, \ldots, l_m$ от корня к соответствующим внешним узлам. И наоборот, если дано множество чисел $l_1, l_2, \ldots, l_m$, всегда ли можно построить расширенное бинарное дерево, в котором эти номера являются длинами пути, расположенными в некотором порядке? Покажите, что это возможно тогда и только тогда, когда $\sum_{j=1}^{m} 2^{-l_j} = 1$.