8. A. A. Shneider, "Classification and analysis of heuristic algorithms for coloring vertices of graphs," Kibernetika, No. 4, 15-22 (1984).
9. Yu. I. Zhuravlev, "Local algorithms for the evaluation of information," Kibernetika, No. 1, 12-19 (1985).
10. Yu. I. Zhuravlev, "Algorithms for the construction of minimal disjoint normal forms for functions of the algebra of the logic," in: Discrete Mathematics and Mathematical Questions of Cybernetics [in Russian], Vol. 1, Nauka, Moscow (1974), pp. 67-98.
11. A. V. Anisimov, "Heuristic algorithms for coloring graphs," in: Seventh All-Union Conference "Problems of Theoretical Cybernetics." Abstracts of Reports [in Russian], Irkutsk (1985), pp. 10-11.
12. A. P. Ershov and G. I. Kozhukhin, "Estimates of the chromatic number of connected graphs," Dokl. Akad. Nauk SSSR, 142, No. 2, 270-273 (1962).
13. A. A. Bul'onkova, "Approximate algorithm for coloring large graphs," in: Problems of Theoretical and Systematic Programming [in Russian], NGU, Novosibirsk (1982), pp. 81-86.
14. N. Christophides, Theory of Graphs. Algorithmic Approach [Russian translation], Mir, Moscow (1978).
15. D. Brelaz, "New methods to color the vertices of a graph," Comm. ACM, 22, No. 4, 251-256 (1979).
16. H. Papadimitriu and K. Steiglitz, Combinatorial Optimization [Russian translation], Mir, Moscow (1985).
17. R. E. Tarjan, "Depth first search and linear graph algorithms," SIAM J. Comput., 1, No. 2, 146-160 (1972).
18. S. Owicki and D. Gries, "An axiomatic proof technique for parallel programs," Acta Inform., 6, 319-340 (1976).
19. V. M. Glushkov, Yu. V. Kapitonova, and A. A. Letichevskii, "Theory of data structures and synchronous parallel computations," Kibernetika, No. 6, 2-15 (1976).
20. A. V. Anisimov, "Locally finite properties of data structures and their evaluations," Kibernetika, No. 6, 1-9 (1983).

HUFFMAN TREES AND FIBONACCI NUMBERS

A. B. Vinokur

UDC 519.10

One of the most important structures in programming is binary trees. The concepts of path length and weighted pathlength of a tree are introduced in [1, 2] in the investigation of binary trees. These characteristics play a substantial role in the analysis of the efficiency of algorithms since they are associated directly with the time of execution of the algorithm and the representation of the trees in the electronic computer. The weighted path length is used, in particular, in cases when the tree is utilized as a search procedure. A Huffman algorithm is described in [1] that permits the connection of a binary tree with minimal weighted path length for given probabilities (weights). We call such trees Huffman trees.

Depending on the domain of application the Huffman trees can allow of different interpretations: optimal search procedure in search theory [1], prefix code with minimal mean word length in information theory [3], etc.

Questions of finding and investigating the extremal structures of data used to obtain the upper and lower bounds of the complexity of algorithms play an important part in the analysis of algorithms.

Extremal Huffman trees are examined in this paper, their connection to Fibonacci numbers, which are related to the solution of different mathematical problems [4, 5], is established. In particular, Fibonacci numbers are applied in programming in the analysis of algorithms.

Let us define the concepts to be used.

A binary tree is an oriented, ordered tree, each component of which has the starting power 2 or 0 [6]. A site with starting power 0 is called terminal and with power 2 non-terminal.

Translated from Kibernetika, No. 6, pp. 9-12, November-December, 1986. Original article submitted January 29, 1985.

 0011-4235/86/2206-0692$12.50

We denote an arbitrary binary tree by D and the number of its terminal sites by $|D| = k$. We assume that the terminal sites of a binary tree are numbered from left to right by 1 to k (the terminal sites are encountered in such a sequence for direct, reverse, and end orders of transversal of the binary tree [1]). Henceforth only a binary tree will be understood to be a tree.

A binary tree is called elongated if at least one of any two adjacent sites is terminal. An elongated binary tree is called left-sided (right-sided) if the right (left) site in each pair of adjacent sites is terminal. Left-sided and right-sided trees are referred to the class of homogeneous binary trees [7]. Homogeneous trees possess extremal complexity properties for definite classes of algorithms [8].

We call a binary tree marked if a certain positive integer (site marker) is set in correspondence with each terminal site. When necessary, marks can be ascribed even to non-terminal sites.

Displayed in Fig. 1 are marked left-sided binary trees with six terminal sites (these sites are outlined by squares).

Let Z be an arbitrary non-decreasing sequence of positive integers $z_1, z_2, \ldots, z_k$ (the number of elements in the sequence is denoted by $|Z|$, $|Z| = k$), D is an arbitrary binary tree with $|D| = k$ and $\pi = \begin{pmatrix} 1 & 2 & \ldots & k \\ j_1 & j_2 & \ldots & j_k \end{pmatrix}$ is a rearrangement such that the marker of the j_i-th site is z_i. We denote such a marked binary tree by $\langle D, \pi, Z\rangle$.

Let us define the following functions on the tree D:

$$E(D) = \sum_{i=1}^{k} l_i, \quad h(D) = \max_i l_i,$$

$$T(D, \pi, Z) = \sum_{i=1}^{k} z_i l_{j_i},$$

where ℓ_i is the path length from the root of the tree D to the i-th terminal site (the site level). The function E(D) is called the length of the (outer) path of the tree, h(D) is the tree height, and $T(D, \pi, Z)$ is the weighted path length of the tree [1]. Thus, we have for the trees in Fig. 1: $E(D_1) = E(D_2) = E(D_3) = 20$; $h(D_1) = h(D_2) = h(D_3) = 5$; $T(D_1, \pi_1, Z_1) = 45$, $T(D_2, \pi_2, Z_2) = 38$, $T(D_3, \pi_3, Z_3) = 31$.

The problem of finding a marked tree with minimal weighted path length for a given sequence and the Huffman algorithm of its solution are examined in [1]. The problem is to find a binary tree $H = H(Z)$ (a Huffman tree) and a rearrangement $\gamma = \gamma(Z)$ such that $T(H, \gamma, Z) = \min\limits_{D}\min\limits_{\pi} T(D, \pi, Z)$ for the sequence Z ($|Z| = k$). The Huffman algorithm is the following. At the j-th step ($j = \overline{1, k-1}$) a non-decreasing sequence $Q_{j-1}(Z): p_1, p_2, \ldots, p_{k-j+1}$ is transformed into the non-decreasing sequence $Q_j(Z): q_1, q_2, \ldots, q_{k-j}$ [by definition $Q_0(Z) = Z$]. The sequence $Q_j(Z)$ is obtained as a result of ordering the non-decreasing of the sequence $p_1 + p_2, p_3, \ldots, p_{k-j+1}$. An element q of the sequence $Q_j(Z)$ is called a terminal element if q belongs to the initial sequence Z and non-terminal otherwise. The passage from $Q_{j-1}(Z)$ to $Q_j(Z)$ corresponds to the appearance of a non-terminal site with marker $p_1 + p_2$ in the Huffman tree being constructed, where the sites with markers p_1 and p_2 ($p_1 \leq p_2$) become filials.

For definiteness, we here assume that 1) if one of the elements p_1 and p_2 of the sequence $Q_{j-1}(Z)$ is terminal while the other is non-terminal, then the non-terminal site with non-terminal marker (p_1 or p_2) becomes the left filial of the site with marker $p_1 + p_2$, while the terminal site is the right filial; 2) in the opposite case, the left filial has the marker p_1 and the right the marker p_2.

The structure of the Huffman tree depends on the nature of the initial sequence Z and the methods of ordering the sequence when going from $Q_{j-1}(Z)$ to $Q_j(Z)$. In this connection, there are generally many Huffman trees for an arbitrary sequence Z.

Of special interest is the investigation of elongated Huffman trees. Such trees possess contrasting extremal properties: the minimum of the weighted length $T(H, \gamma, Z)$ for the sequence Z is achieved for a maximum of the ordinary length E(H) and the height h(H). Let us examine what the sequences constructing elongated Huffman trees should be.

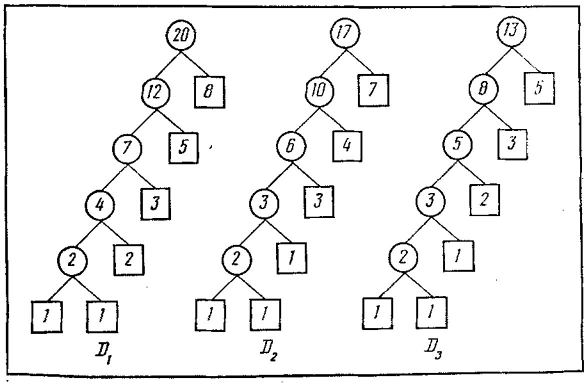

Fig. 1

TABLE 1

Sequence cost	k									
	1	2	3	4	5	6	7	8	9	10
$A \in M_0$	1	1	2	3	5	8	13	21	34	55
$B \in M_1$	1	1	1	3	4	7	11	18	29	47
$C \in M_2$	1	1	1	2	3	5	8	13	21	34
$S_0(HL, k)$	—	2	6	13	25	45	78	132	220	363
$S_1(HL, k)$	—	2	5	11	21	38	66	112	187	309
$S_2(HL, k)$	—	2	5	10	18	31	52	86	141	230

We introduce a number of concepts and notations. As before, Z is a non-decreasing sequence of positive integers $z_1, z_2, \ldots, z_k$, $Q_{j-1}(Z)$ and $Q_j(Z)$ are non-decreasing sequences in the j-th stage of the Huffman algorithm.

Definition 1. A sequence Z is called strict (or strict in value) if $q_2 \neq q_3$ (i.e., $q_2 < q_3$) in each sequence $Q_j(Z)$ $(j = \overline{1, k-3})$.

Definition 2. A strict sequence Z is called absolutely strict (or strict in address) if $z_2 \neq z_3$ (i.e., $z_2 < z_3$).

Definition 3. A sequence Z is called non-strict if there is a sequence $Q_j(Z)$ $(1 \leq j \leq k-3)$ in which $q_2 = q_3$.

Definition 4. A non-strict sequence Z is called alternative (or absolutely non-strict) if there is a $q_2 = q_3$ in every sequence $Q_j(Z)$ $(j = \overline{1, k-3})$ and a $z_2 = z_3$ in the sequence Z.

It follows from Definitions 1 and 3 that any sequence Z is either strict or non-strict. If the sequence is strict here, then in each stage of the Huffman algorithm except perhaps the first (and also in the first stage for an absolutely strict sequence), the selection of two minimal elements is realized single-valuedly. If the sequence is non-strict, then in certain stages of the Huffman algorithm (and in all stages for an alternative sequence), the selection of two minimal elements can be realized by more than one method. Therefore, for a strict sequence there exists a unique Huffman tree, and for a non-strict sequence more than one such tree. The reverse assertion is also true: if a unique Huffman tree exists for a sequence, then the sequence is strict, if more than one tree exists then the sequence is non-strict.

Let M_0 (M_1, respectively) denote the set of such absolutely strict (simply strict, respectively) sequences for which the Huffman tree is elongated, M_2 is the set of all (strict and non-strict) sequences for which an elongated Huffman tree exists. As follows from the method described above for forming sites, if the Huffman tree is elongated, then it will be left-sided. We denote left-sided Huffman trees by HL, thereby distinguishing them from the Huffman trees H of arbitrary structure. Evidently $M_0 \subset M_1 \subset M_2$. If the tree H is a Huffman tree for one sequence $Z: z_1, z_2, \ldots, z_k$, then it is a Huffman tree for an infinite (countable) number of sequences, for instance, for sequences of the form $Z^{(m)}$ in which $z_i^{(m)} = m z_i$.

Definition 5. The sequence $Z_{\min}$ $(|Z_{\min}| = k)$ is called a minimizing sequence of the tree H $(|H| = k)$ in the class M if H is a Huffman tree H(Z) for all $Z \in M$ $(|Z| = k)$ and $T(H, \pi_{\min}, Z_{\min}) = \min\limits_{Z \in M} \min\limits_{\pi} T(H, \pi, Z)$, where $\pi_{\min}$ is a certain rearrangement. The quantity $S_M(H, k) = T(H, \pi_{\min}, Z_{\min})$ is called the cost of the tree H in the class of sequences M.

The cost of a tree is the lower (allowable) bound for the weighted path length for this tree.

Let us consider the problem of finding minimizing sequences and cost of an elongated (left-sided) tree HL in the classes M_0, M_1, and M_2. The costs S_{M_0}, S_{M_1}, and S_{M_2} will be denoted by S_0, S_1, and S_2. We denote the Fibonacci sequence $u_1 = 1$, $u_2 = 1$, $u_j = u_{j-1} + u_{j-2}$ $(j \geqslant 3)$ [5] by $U: u_1, u_2, \ldots, u_k$. Thus, for instance, $u_3 = 2$, $u_4 = 3$, $u_5 = 5$, $u_6 = 8$, $u_7 = 13$, $u_8 = 21$, etc. Furthermore, for an arbitrary sequence $Z: z_1, z_2, \ldots, z_k$ we use the notation $\mathrm{sum}(z_k) = \sum_{i=1}^{k} z_i$.

THEOREM 1. A Fibonacci sequence is a minimizing sequence of a left-sided binary tree HL ($|HL| = k$) in the class M_0. The cost is $S_0(HL, k) = u_{k+4} - (k+4)$.

Proof. Let $A: a_1, a_2, \ldots, a_k\ (a_i \leqslant a_{j+1})$ be a minimizing sequence of the tree HL in the class M_0. Let us determine the constraint which this sequence should satisfy. In the first stage of the Huffman algorithm, terminal elements a_1 and a_2 are selected (as minimal) in the sequence $Q_0(A) = A$. Since A is a minimizing sequence, then each of its elements should be minimally possible, in particular, $a_1 = a_2 = 1$. Moreover, if A is an absolutely strict sequence (i.e., $a_2 < a_3$), then $a_3 = a_2 + 1 = a_1 + 1$.

A left-sided tree should certainly be obtained as a result of applying the Huffman algorithm to the sequence A. In each stage (except the first, naturally) one element in the pair of selected minimal elements of the sequence $Q_j(A)$ should be terminal (i.e., should belong to the sequence A), and the other non-terminal. In particular, the non-terminal element $a_1 + a_2$ [the unique non-terminal element in $Q_1(A)$] should be selected simultaneously with the terminal element a_3 (which will evidently certainly be selected) in the second stage. Hence, taking into account that $a_3 = a_1 + a_2$ and A is an absolutely strict sequence, we obtain $a_1 + a_2 < a_4$. The minimally possible value for a_4 is $a_4 = a_1 + a_2 + 1 = \mathrm{sum}(a_2) + 1$. Reasoning analogously for the j-th stage of the algorithm, we obtain that $Q_j(A)$ is the sequence $a_{j+2}, \mathrm{sum}(a_{j+1}), a_{j+3}, \ldots, a_k\ (j = \overline{1, k-2})$. Hence, taking into account that $a_3 = a_1 + 1$, we have $\mathrm{sum}(a_{j+1}) < a_{j+3}$ and $a_{j+3} = \mathrm{sum}(a_{j+1}) + 1\ (j = \overline{0, k-3})$. Considering the difference $a_{j+3} - a_{j+2} = (\mathrm{sum}(a_{j+1}) + 1) - (\mathrm{sum}(a_j) + 1) = \mathrm{sum}(a_{j+1}) - \mathrm{sum}(a_j) = a_{j+1}$, we obtain $a_{j+3} = a_{j+2} + a_{j+1}\ (j = \overline{1, k-3})$. Therefore, taking into account that $a_1 = a_2 = 1$ and $a_3 = a_2 + a_1$, the sequence A is the Fibonacci sequence $U: u_1, u_2, \ldots, u_k$.

Therefore, the left-sided tree HL is a Huffman tree for the Fibonacci sequence, i.e.,

$$T(HL, \pi_0, U) = \min_D \min_\pi T(D, \pi, U),$$

where, as follows from the construction of the tree by the Huffman algorithm, π_0 is the identity rearrangement (see [1] for its existence for the arbitrary sequence Z).

Let us now determine the cost $S_0(HL, k) = T(HL, \pi_0, U) = \sum_{i=1}^{k} u_i l_{j_i}$. Since π_0 is the identity rearrangement ($j_i = 1$), then $S_0(HL, k) = \sum_{i=1}^{k} u_i l_i$. Furthermore, taking into account that $|HL| = k$ and therefore $h(HL) = k - 1$, we obtain $\ell_i = k + 1 - i\ (i = \overline{2, k})$, $\ell_1 = k - 1$, then $S_0(HL, k) = \sum_{i=1}^{k} \sum_{j=1}^{i} u_j - u_1 = \sum_{i=1}^{k} \mathrm{sum}(u_i) - 1$. Since $(u_i) = u_{i+2} - 1$ [5], then $S_0(HL, k) = \sum_{i=1}^{k} (u_{i+2} - 1) - 1 = \sum_{i=1}^{k} u_{i+2} - (k+1) = \sum_{i=3}^{k+2} u_i - (k+1) = \mathrm{sum}(u_{k+2}) - (k+3) = u_{k+4} - (k+4)$.

The theorem is proved.

THEOREM 2. A sequence $B: b_1, b_2, \ldots, b_k$ such that $b_1 = b_2 = b_3 = 1$, $b_i = u_i - u_{i-4}\ (i \geqslant 4, u_0 = 0)$. is a minimizing sequence of a left-sided binary tree HL ($|HL| = k$) in the class M_1. The cost is $S_1(HL, k) = u_{k+4} - u_k - (k+3)$.

Proof. Taking into account that B is a strict (not certainly absolute strict) sequence, and reasoning analogously to the proof of the previous theorem, we obtain $b_1 = b_2 = 1$, $b_3 = b_2 = 1$, $b_{j+3} = \mathrm{sum}(b_{j+1}) + 1\ (j = \overline{1, k-3})$ and therefore $b_{j+3} = b_{j+2} + b_{j+1}\ (j = \overline{2, k-3})$. We show by induction that $b_i = u_i - u_{i-4}$ for $i \geqq 4$. The initial step in the induction is ($i = 4$, $i = 5$): $b_4 = b_1 + b_2 + 1 = 3 = u_4 - u_0$, $b_5 = b_4 + b_3 = 4 = u_5 - u_1$. We now assume that $b_i = u_i - u_{i-4}$ for $4 \leqslant i \leqslant m$, then we have $b_{m+1} = b_m + b_{m-1} = (u_m - u_{m-4}) + (u_{m-1} - u_{m-5}) = (u_m + u_{m-1}) - (u_{m-4} + u_{m-5}) = u_{m+1} - u_{m-3}$. The minimizing sequence B is defined. The principle of calculating the cost $S_1(HL, k)$ is analogous to the calculation of $S_0(HL, k)$. Consequently, we obtain $S_1(HL, k) = u_{k+4} - u_k - (k+3)$.

THEOREM 3. A sequence $C: c_1, c_2, \ldots, c_k$ such that $c_1 = 1$, $c_i = u_{i-1}\ (i \geqslant 2)$ is a minimizing sequence of the left-sided binary tree HL ($|HL| = k$) in the class M_2. The cost is $S_2(HL, k) = u_{k+3} - 3$.

Proof. Taking into account that C is non-strict, and reasoning appropriately, we obtain $c_1 = c_2 = 1$, $c_3 = c_2 = 1$, $c_{j+3} = \mathrm{sum}(c_{j+1})\ (j = \overline{0, k-3})$ and $c_{j+3} = c_{j+2} + c_{j+1}\ (j = \overline{1, k-3})$. Since $c_2 = c_3 = 1$, then $c_i = u_{i-1}$ ($i \geq 2$). Therefore, the minimizing sequence C is a Fibonacci sequence "shifted" one element to the right. Evaluating the cost, we obtain $S_2(HL, k) = u_{k+3} - 3$.

COROLLARY 1. The non-strict sequence C is alternative.

This follows from the fact that in each stage j ($j = \overline{2, k-2}$) there are exactly two variants of the selection of a pair of minimal elements: 1) terminal and non-terminal elements; 2) both elements are either terminal or non-terminal.

COROLLARY 2. For the alternative sequence C ($|C| = k \geq 3$) there exist 2^{k-3} different marked Huffman trees (one of them is left-sided).

The trees $\langle HL, \pi_0, A\rangle$, $\langle HL, \pi_0, B\rangle$, and $\langle HL, \pi_0, C\rangle$ (D_1-D_3, respectively) are displayed in Fig. 1 for $|HL| = 6$. Let us note that even among the elongated Huffman trees, the trees $\langle HL, \pi_0, A\rangle$, $\langle HL, \pi_0, B\rangle$, and $\langle HL, \pi_0, C\rangle$ are the most contrasting. Thus, C is the minimal of the sequences Z for which the minimum of the weighted path length of the tree is achieved on a tree of maximal height and pathlength of the tree.

Comparing the costs obtained for the left-sided tree, we obtain $S_0(HL, k) - S_1(HL, k) = u_k - 1$, $S_1(HL, k) - S_2(HL, k) = u_{k+1} - k$, i.e., as should have been assumed $S_2(HL, k) \leq S_1(HL, k) \leq S_0(HL, k)$ (the inequalities are strict for $k > 3$). Therefore, the sequence C is an absolute minimizing sequence of the left-sided tree and $S_2(HL, k)$ is the least weighted path length among all the sequences.

Presented in Table 1 are values of the elements of the sequences A, B, C and the corresponding costs of a left-sided tree for certain values of k.

Let us examine the asymptotic behavior of the cost of a left-sided tree. Taking into account that $u_k \approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^k$ [1] and using the notation $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.6180$, we obtain

$$S_0(HL, k) \sim \frac{1}{\sqrt{5}}\varphi^4\varphi^k \approx 2.96\cdot(1.62)^k,$$

$$S_1(HL, k) \sim \frac{1}{\sqrt{5}}(\varphi^4 - 1)\varphi^k \approx 2.53\cdot(1.62)^k,$$

$$S_2(HL, k) \sim \frac{1}{\sqrt{5}}\varphi^3\varphi^k \approx 1.83\cdot(1.62)^k,$$

i.e., the costs depend exponentially on the number of terminal sites of the tree, which differs here by just constant factors.

LITERATURE CITED

1. D. Knute, Art of Programming for Electronic Computers. Fundamental Algorithms [Russian translation], Vol. 1, Mir, Moscow (1976).
2. E. Rheingold, U. Nivergelt, and N. Deo, Combinatorial Algorithms. Theory and Practice [Russian translation], Mir, Moscow (1980).
3. A. Renyi, On the Mathematical Theory of Trees. Trilogy on Mathematics [Russian translation], Mir, Moscow (1980), pp. 353-375.
4. A. Renyi, Variations on a Theme of Fibonacci. Trilogy on Mathematics [Russian translation], Mir, Moscow (1980), pp. 326-352.
5. N. N. Vorob'ev, Fibonacci Numbers [in Russian], Nauka, Moscow (1984).
6. A. Berztiss, Data Structure [in Russian], Statistika (1974).
7. A. B. Vinokur and G. P. Kozhevnikova, "On a class of regular trees used in the complexity analysis of algorithms," Preprint No. 82-18, Inst. Kibern., Akad. Nauk, Ukr. SSR, Kiev (1982).
8. A. B. Vinokur and G. P. Kozhevnikova, "Approach to the formalization of an estimate of the efficiency of programmed transformations on the basis of an analysis of the properties of algorithms and data structures," Optimization and Transformation of Programs: Materials of an All-Union Seminar (Novosibirsk, 1983) [in Russian], Vol. 1, 67-75, Vychisl. Tsentr. Sib. Otd. Akad. Nauk SSSR, Novosibirsk (1983).