ESTIMATE OF COMPUTER SORTING BY THE SHELL METHOD

S. I. Kozlova and V. N. Mikhailov

UDC 681.3

In [1] D. L. Shell presented a highly effective method of computer sorting. Various salient features of this method were discussed in [2, 3]. In [4], four variants of the Shell sort were investigated by numerical simulation.

In the present paper we obtain exact analytic formulas for the mean number of comparisons and exchanges it is necessary to perform when using one of the variants of the Shell method to sort an unordered array with an arbitrary number of entries. Computational results are provided.

THE SHELL METHOD

We consider the following variant of the Shell sort. The sequence of elements

\[ a_1, a_2, \ldots, a_N \]

is ordered in several steps. In the first step, elements \( a_i \) and \( a_i + m_1 \) are ordered independently of one another, where

\[ m_i = 2^M < N \leq 2^{M+1}, \quad i = 1, 2, \ldots, N - m_1. \]

In the second step, sequences of four elements are ordered:

\[ a_i, a_i + m_2, a_i + 2m_2, a_i + 3m_2, \quad m_2 = 2^{M-1}, \]

\[ i = 1, 2, \ldots, m_2. \]

If \( N < 2^{M+1} \) then certain subsequences consist of only three elements. On the \( k \)-th step one sorts the elements

\[ a_i, a_i + m_k, a_i + 2m_k, a_i + 3m_k, \ldots, a_i + n_k m_k, \]

\[ m_k = 2^{M-k+1}, \quad i = 1, 2, \ldots, m_k, \]

where \( n_k \) is the largest integer satisfying the inequality \( i + n_k m_k \leq N \). On the final, \( (M+1) \)-th step, the entire sequence is ordered.

Let us compare the subsequences which are sorted on the \((k-1)\)-st and \( k \)-th steps. On the \((k-1)\)-st step we sort the subsequences

\[ a_i, a_{i+m_{k-1}}, a_{i+2m_{k-1}}, \ldots, a_{i+n_{k-1} m_{k-1}}. \]

(2)

We note that the elements

\[ a_i, a_{i+2m_k}, a_{i+3m_k}, \ldots \]

(3)

for \( i = 1, 2, \ldots, m_k \) coincide with subsequence (2) for \( i = 1, 2, \ldots, m_k = m_{k-1}/2 \), since \( 2m_k = m_{k-1} \), \( 4m_k = 2m_{k-1} \), etc. Exactly the same, the elements

\[ a_{i+m_k}, a_{i+3m_k}, \ldots \]

(4)

of subsequences (1) coincide, for \( i = 1, 2, \ldots, m_k \), with subsequences (2) when \( i = m_k + 1, m_k + 2, \ldots, m_k - 1 \). But, the subsequences of (2) are already ordered after the \((k-1)\)-th step, so that, prior to the beginning of the \(k\)-th step, subsequences (2) and (3) of sequence (1) will already have been sorted. Consequently, in the variant of the Shell sort under consideration here, we sort sequences of the form

\[
\begin{align*}
\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_n, \beta_n, \\
\alpha_1 < \alpha_2 < \ldots < \alpha_n, \quad \beta_1 < \beta_2 < \ldots < \beta_n.
\end{align*}
\]

For subsequences of (1), at each step we use a method of exchange with replacement [5] so that, to define the effectiveness of the Shell method, it is necessary to first find the mean numbers of comparisons and exchanges which are required to sort sequence (5) by the method of exchange with replacement.

In what follows we shall call sequence (5) a semisorted sequence.

**SORTING A SEMISORTED SEQUENCE**

Assume that sequence (5) is ordered. Any \( n \) elements, selected from this sequence in increasing order, can form a subsequence \( \alpha_1, \alpha_2, \ldots, \alpha_n \) of unsorted sequence (5), while the remaining \( n \) elements of the ordered sequence form a second subsequence \( \beta_1, \beta_2, \ldots, \beta_n \). It follows from this that one can form \( C_n^m \) different sequences (5) from the 2n elements. We shall consider all these sequences to be equally probable.

Let us find the number of all the comparisons and exchanges necessary to sort the elements \( \alpha_k \) and \( \beta_k \), \( k = 1, 2, \ldots, n \) in all \( C_n^m \) sequences of (5).

We put into correspondence with elements \( \alpha_k \) and \( \beta_k \) the natural numbers \( i_k \) and \( j_k \) which show which positions must be occupied by these elements in the ordered sequence (5). Subsequences \( \alpha_1, \alpha_2, \ldots, \alpha_n \) and \( \beta_1, \beta_2, \ldots, \beta_n \) are ordered, so that the following inequalities must hold:

\[
\begin{align*}
i_1 < i_2 < \ldots < i_n, \\
j_1 < j_2 < \ldots < j_n.
\end{align*}
\]

Since \( i_k \) and \( j_k \) are natural numbers, it follows from these inequalities that

\[
k \leq i_k \leq k + n, \quad k \leq j_k \leq k + n.
\]

Sorting of sequence (5) is equivalent to the sorting of the natural number sequence

\[
i_1, i_2, i_3, \ldots, i_n,
\]

if we define only the operation of comparison on them, so that henceforth we shall speak of sequence (8).

According to the method of exchange with replacement [5], element \( i_k \), via successive comparisons with the elements to its left, is placed into the already ordered subsequence of numbers \( i_1, i_2, \ldots, i_{k-1}, i_k \).

Let \( i_k = k + l \). Thanks to the inequalities in (7), the quantity \( l \) can have the values 0, 1, 2, \ldots, \( n \). We note now that, of the numbers 1, 2, \ldots, \( k + l \), \( k \) numbers go into the formation of the sequence \( i_1, i_2, \ldots, i_{k-1}, i_k \), while the remaining \( l \) numbers can enter only into the construction of sequence \( j_1, j_2, \ldots, j_l \), since the elements \( i_{k+1}, i_{k+2}, \ldots, i_n \) cannot assume these values due to the inequalities in (6). Therefore, if \( l = k - 1 \) then the subsequence \( j_1, j_2, \ldots, j_{k-1} \) is formed of the numbers less than \( i_k \) and, consequently, for the ordering of element \( i_k \) only one comparison is necessary and not one exchange. If \( l < k - 1 \) then the elements \( j_{l+1}, j_{l+2}, \ldots, j_{k-1} \) must be larger than number \( i_k \). In this case it is necessary to perform \( (k-l) \) comparisons and \( (k-l-1) \) exchanges.

Let us find the number \( R_{k,l} \) of sequences in which \( i_k = k + l \). The sequence \( i_1, i_2, \ldots, i_{k-1} \) can be chosen by \( C_{k-l-1}^{k-1} \) methods. The subsequence \( i_{k+1}, i_{k+2}, \ldots, i_n \) can be constructed only from the numbers \( k + l + 1, k + l + 2, \ldots, 2n \); there are \( C_n^{2n-k-l} \) such possibilities, so that we have

\[
R_{k,l} = C_{k-l-1}^{k-1} C_n^{2n-k-l}.
\]

*The Russian notation for the combinations of \( t \) things taken \( r \) at a time is \( C_t^r \), not \( C_t^r \) as is customary here - Translator.*