SIGNIFICANT IMPROVEMENTS TO THE FORD-JOHNSON ALGORITHM FOR SORTING

T. D. BUI and MAI THANH
Department of Computer Science, Concordia University, 1455 De Maisonneuve Boulevard West, Montreal, Québec, Canada H3G 1M8

Abstract.

For almost twenty years, the Ford-Johnson algorithm for sorting $t$ items using comparisons was believed to be optimal. Recently, Manacher was able to show that the Ford-Johnson algorithm is not optimal for certain ranges of values of $t$. In this paper, we present some new algorithms which achieve much stronger results compared to Manacher’s algorithms.


1. Introduction.

This paper deals with sorting by comparisons. In particular, we are interested in the worst-case analysis. That is, we ask the question: What is the minimum number of comparisons which is sufficient to sort $t$ items in all cases?

In 1959, Ford and Johnson [3] published a paper entitled: “A tournament problem” in which they introduced an algorithm for sorting, later known as the Ford-Johnson algorithm (we denote this algorithm by FJA). Briefly speaking, the FJA has three steps:

Step 1. Divide the elements in $\lfloor t/2 \rfloor$ pairs and make pairwise comparisons. Leave one element out if $t$ is odd.

\[
\begin{array}{cccccc}
& & & \cdot & & \\
& & & & & \\
& & \cdot & & & \\
& & & & & \\
& & & \cdot & & \\
\end{array}
\]  
(if $t$ is odd)

Fig. 1.

Received September 1983. Revised September 1984.
SIGNIFICANT IMPROVEMENTS TO THE FORD-JOHNSON ALGORITHM FOR SORTING

Step 2. Sort the \( \lfloor t/2 \rfloor \) larger elements found in Step 1 by the FJA.

\[ \begin{align*}
& a_1 \quad a_2 \quad a_3 \\
& b_1 \quad b_2 \quad b_3 \quad (\text{if } t \text{ is odd})
\end{align*} \]

Fig. 2.

Step 3. Insert the \( b \)'s elements in sequences into the main chain:
\( (b_1, a_1, a_2, \ldots, a_{\lfloor t/2 \rfloor}) \) which is already sorted.

The sequences to be inserted are:
\( (b_3, b_2), (b_5, b_4), (b_1, b_1, \ldots, b_{5}), \ldots, (b_i, b_i-1, \ldots, b_{i-1}+1), \ldots \)

The last sequence may not be a complete set as defined.

\[ \begin{align*}
& a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \\
& b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5 \quad (\text{if } t \text{ is odd})
\end{align*} \]

Fig. 3.

Let \( F(t) \) be the number of comparisons required by the FJA to sort \( t \) items. Then (see [5]):

\[ (1) \quad F(t) = \lfloor \log_2 3t \rfloor - \lfloor \frac{1}{3} \log_2 6t \rfloor + \lfloor \frac{1}{2} \log_2 6t \rfloor. \]

If we substitute \( t = (4/3)^{2^k + f} \) where \( k \) is an integer and \( f \) is a real number \( 0 \leq f \leq 1 \), then

\[ (2) \quad F(t) = t \log_2 t - c(t)t + \frac{1}{2} \log_2 t + O(1) \]

where \( c(t) = -1 + f + 2^{-1-f} - \log_2 3. \)

For \( t = u_k = \lfloor (4/3)^{2k} \rfloor \), i.e. \( f = 0 \) or 1, the coefficient \( c(t) \) has minimum value of \(-1.415\) which is remarkably close to the absolute lower bound of \(-1.443\). The information-theoretic lower bound is given by [5]:

\[ (4) \quad E(t) = \lceil \log_2 t \rceil = t \log_2 t - 1.443t + O(\log_2 t). \]

For this reason, for almost twenty years, the FJA has not been improved.