JOURNAL OF ALGORITHMS 11, 242–251 (1990)

Tight Lower Bounds for Shellsort*

MARK ALLEN WEISS

School of Computer Science, Florida International University, University Park, Miami, Florida 33199

AND

ROBERT SEDGEWICK

Department of Computer Science, Princeton University, Princeton, New Jersey 08540

Received March 12, 1988; revised August 29, 1989

It is proved that the running time of Shellsort using an increment sequence given by Sedgewick is $\Omega(N^{4/3})$ which matches the known upper bound. Extending this proof technique to various increment sequences leads to lower bounds that in general always match the known upper bounds. This suggests that Shellsort runs in $\Omega(N^{1+\varepsilon/\sqrt{\log N}})$ for increment sequences of practical interest and that no increment sequence exists that would make Shellsort optimal. © 1990 Academic Press, Inc.

1. INTRODUCTION

Shellsort is a simple sorting algorithm proposed by D. Shell [16] in 1959. For nearly sorted or mid-sized files (a few thousand elements), Shellsort performs as well as or better than any known algorithm, including quicksort. Furthermore, it is an in-place sorting algorithm requiring little extra space and is easy to code.

Shellsort uses a sequence of integers $h_t, h_{t-1}, \ldots, h_1$ and works by performing insertion sort on subfiles consisting of elements h_i apart. We will call one of these sorting operations an h_i-sort. In an h_i-sort, an element in position p is placed in its correct order in its subfile by comparing it against elements in positions $p - h_i$, $p - 2h_i$, etc.

Shellsort works by performing passes consisting of an h_t-*sort*, h_{t-1}-*sort*, and so on, concluding with an $h_1 = 1$-*sort*. It is both necessary and sufficient that some pass do a 1-*sort* in order for the algorithm to be

*This work was supported by National Science Foundation Grant DCR-8605962.

0196-6774/90 $3.00

guaranteed to sort a file. An important property of Shellsort is that if a *k-sorted* file is subsequently *h-sorted*, the file remains *k-sorted* [5, 8, 11]. This is the property that makes Shellsort fast. As insertion sort works well for files that are nearly sorted, we expect that it might be fairly inexpensive to h_i*-sort* a file if it has already been h_{i+1}-, h_{i+2}-, ..., h_t*-sorted*.

Typically, the increment sequences used are "almost" geometric sequences with $h_k = O(\alpha^k)$ for some α, concluding with h_t being the largest integer in this sequence less than N. This is by no means a requirement; however, empirically these increment sequences perform better than others. For increments $1, 2, \ldots, h_k = 2^k, \ldots$, originally proposed by Shell, Shellsort is quadratic in the worst case, and $O(N^{3/2})$ on average [8]. At the other end of the spectrum, Pratt [11] gives a set of $O(\log^2 N)$ increments, for which the running time is $\Theta(N \log^2 N)$. This is the best known bound for Shellsort. It performs poorly in practice unless N is unrealistically large because this approach yields too many increments. Between these extremes, new results have lowered the worst-case running time of Shellsort to values not quite optimal, but considerably better than quadratic. On the other hand not even the asymptotic growth of the average case performance is known for the types of sequences used in practice, although none seem to be $O(N \log N)$.

In this paper, we consider lower bounds on the worst-case running time. Pratt showed that for increment sequences of the form $1, \ldots, h_k = c_1\alpha^k + c_2, \ldots, \alpha$ an integer, Shellsort runs in $\Theta(N^{3/2})$ (subject to certain technical conditions). This property is held by most of the increment sequences that have been tried. However Sedgewick [12] showed that if $h_k = 4 \cdot 4^k + 3 \cdot 2^k + 1$, then the running time is $O(N^{4/3})$. Our first main result in this paper is to prove this bound is tight, by constructing a permutation that takes the required time to sort. Incerpi and Sedgewick [6] have extended this result by providing an increment sequence which gives a running time of $O(N^{1+\varepsilon/\sqrt{\log N}})$. Our second main result is to show that this bound is also tight, under the assumption that an unproven (but rather fundamental) conjecture is true. Moreover, it appears that if the increments are of the form $h_k = \Theta(\alpha^k)$, then the bound of Incerpi and Sedgewick is the best possible.

Cypher [2] has recently shown an $\Omega(N \log^2 N/\log\log N)$ lower bound for the size of Shellsort-based sorting networks for monotonic increment sequences, which seems to imply the same bound for the sequential Shellsort algorithm. Although our lower bound is for a less general class of increment sequences, this class covers virtually all of the increment sequences proposed so far. Furthermore, our bound is much larger (and tight).

Section 2 briefly reviews the methods used to obtain the aforementioned upper bounds. In Section 3, we discuss the Frobenius pattern, and prove a

lemma about the number of inversions in this pattern. We use this lemma to prove the lower bound. In Section 4, we discuss generalizations of this result to other increment sequences. Open problems are discussed in Section 5.

2. Previous Upper Bounds

To derive our bounds for Shellsort, we consider an old problem from number theory:

Suppose that a country wishes to issue only k different types of stamps. What is the largest postage that cannot be placed exactly on an (infinite-sized) envelope?

This is known as the Frobenius problem, apparently because the mathematician Frobenius mentioned it often in his classes [1]. A more formal definition follows:

Definition. The *Frobenius number*, $g(a_1, a_2, \ldots, a_k) \equiv$ the largest integer which cannot be represented as a linear combination with non-negative, integer coefficients, of $a_1, a_2, \ldots, a_k$.

The function $g(\ )$ is called the *Frobenius function*. We make several simple observations: First, we assume $a_1 < a_2 < \cdots < a_k$ without loss of generality. Throughout the rest of this paper, we shall make this assumption. Now, $g(1, \ldots) = 0$, as all positive integers are representable; we thus assume $a_1 > 1$. Also, we may assume that each a_i is independent of the other arguments (that is, it cannot be represented as a linear combination of the other arguments), since otherwise it could be removed without affecting the result. Finally, $g(a_1, a_2, \ldots, a_k)$ is defined iff $gcd(a_1, a_2, \ldots, a_k) = 1$.

For $k = 2$, the solution (due to Sharp [15]) is:

$$g(a_1, a_2) = (a_1 - 1)(a_2 - 1) - 1 \tag{2.1}$$

provided, of course, that a_1 and a_2 are relatively prime. A formula for $k = 3$ is known [13], but is fairly complicated and no general solution is known for $k > 3$. Johnson [7] has shown that

$$g(a_1, a_2, \ldots, a_k) = d \cdot g\left(\frac{a_1}{d}, \frac{a_2}{d}, \ldots, \frac{a_{k-1}}{d}, a_k\right) + (d - 1)a_k, \tag{2.2}$$

where

$$d = gcd(a_1, a_2, \ldots, a_{k-1}).$$

Incerpi and Sedgewick [6] provided a lower bound for the Frobenius function:

$$g(a_1, a_2, \ldots, a_k) \geq \Omega\left(a_1^{(1+1/(k-1))}\right), \tag{2.3}$$

provided of course that $a_1 < a_2 < \cdots < a_k$.

Equations (2.1), (2.2) and the result for the case $k = 3$ are sufficient to prove all the known upper bounds for Shellsort. The main result is that if the increment sequence satisfies

$$g(h_i, h_{i+1}, \ldots, h_t) = O(h_i^c)$$

then a time bound of $O(N^{2-1/c})$ can always be proved [6, 10, 12, 17].

3. A New Lower Bound

In this section, we derive the main results of this paper by presenting a permutation that is asymptotically as bad as possible for Shellsort.

Definition. Given a file of integers represented by $x_1, x_2, \ldots, x_N$, an inversion is any pair (i, j) such that $i < j$ and $x_i > x_j$.

In the worst case, a file in reverse order can have $\Theta(N^2)$ inversions. A file has no inversions iff it is sorted, and exchanging two adjacent elements that are out of sequence removes exactly one inversion.

To facilitate our calculations, we will make the simplifying assumption that there are only two keys, 0 and 1 in the file to be sorted. The following lemma then applies:

3.1. Lemma (Swapping lemma). *Swapping a* 0 *and a* 1 *a distance d apart in a* 0–1 *permutation removes exactly d inversions.*

Proof. The proof is simple and is omitted. □

Remark. We can easily prove that if the elements of the permutation are not restricted to be 0 and 1, then the number of inversions removed lies between 1 and $2d - 1$.

The general idea of the proof is to construct a permutation with $\Omega(N^2)$ inversions and show that no exchange can remove too many inversions—thus, many exchanges are required. Using 0–1 permutations instead of general permutations could only affect the constant in the time bound (because of the remark above). Also, although we prove our theorem only for a few specific permutation sizes, it is clear that we can always pad the input, again affecting only the constant in the time bound. Thus, neither of these restrictions is significant.

The natural permutation to consider is a file in reverse order. This "natural" permutation is not a bad one for Shellsort (Shellsort runs in $O(N \log N)$ for this permutation [18]), because the early passes quickly

bring sortedness to the file. What we need is a permutation that is initially very unsorted and that is not made nearly-sorted by early passes.

The permutation we will use is closely related to the specific increment sequence and the Frobenius function. In particular, we have $h_1 = 1$ and $h_k = 4^{k-1} + 3 \cdot 2^{k-2} + 1$ for $k > 1$. For any value of $k > 1$, we choose $N_k = g(h_k, h_{k+1}, \ldots, h_\infty) + 1$. Eventually, there will be some maximum $h_t < N_k$, and thus we may write $N_k = g(h_k, h_{k+1}, \ldots, h_t) + 1$. If we store our permutation P_k as $p_0, p_1, \ldots, p_{N-1}$, then we define P_k as follows:

DEFINITION. $p_i \equiv 1$ iff i is representable as a linear combination in non-negative integer coefficients of $h_k, h_{k+1}, \ldots, h_t$, and 0 otherwise.

Remark. $p_0 = 1$ by the above definition. We will use the term *Frobenius pattern* to describe P (for obvious reasons). We will say that the *index* of any element p_i is i.

Our permutation has the following very desirable property:

3.2. LEMMA. *No exchanges are performed by Shellsort for the increments* $h_t, h_{t-1}, \ldots, h_k$ *on permutation* P_k.

Proof. For any $h_{t'}$ such that $k \le t' \le t$, if $a_x = 1$, then $a_{x+h_{t'}}$ must also equal 1, so that the lemma follows. □

This lemma shows us that the early passes do no sorting work at all for our permutation. We now show that P_k has a lot of inversions to start with, so that we can expect Shellsort to run slowly on it. We need to estimate the number of inversions in our permutation. We start with the following lemma:

3.3. LEMMA. $N_k = \Omega(h_k^{3/2})$.

Proof. We have (for $k > 1$)

$$\begin{aligned} h_k &= 4^{k-1} + 3 \cdot 2^{k-2} + 1 \\ h_{k+1} &= 4 \cdot 4^{k-1} + 6 \cdot 2^{k-2} + 1 \\ h_{k+2} &= 16 \cdot 4^{k-1} + 12 \cdot 2^{k-2} + 1 \\ &\cdots \end{aligned}$$

Partition the permutation into lines, such that line l contains $p_{(l-1)h_k}$ to p_{lh_k-1}. Each line contains more than 4^{k-1} elements. Consider a 1 on line $\alpha = 2^{k-3}$. Its index clearly must have the form

$$\alpha \cdot 4^{k-1} + 3\beta \cdot 2^{k-2} + \gamma.$$

Suppose that some element on line $\alpha = 2^{k-3}$, with index $ph_k + qh_{k+1} + rh_{k+2} + sh_{k+3} + \ldots$, is 1. Then the only possibility is

$$\begin{aligned} p + 4q + 16r + 64s + 256t + \cdots &= \alpha \\ p + 2q + 4r + 8s + 16t + \cdots &= \beta \\ p + q + r + s + t + \cdots &= \gamma. \end{aligned}$$

These three equations imply that $0 \le \gamma \le \beta \le \alpha$. Thus the number of ones on line α is $|(\beta, \gamma)| \le \alpha^2$, where $|(x_1, x_2, \ldots, x_k)|$ denotes the number of ordered *k-tuples* $(x_1, x_2, \ldots, x_k)$. It follows that line $\alpha = 2^{k-3}$ has at most 4^{k-3} ones. This implies that there are at least $15 \cdot 4^{k-3}$ zeros on this line, and that the Frobenius pattern does not end prior to this line (since this line is not all ones). Since we have $\Omega(2^k)$ lines containing $\Omega(4^k)$ elements per line, the total number of elements is $\Omega(8^k) = \Omega(h_k^{3/2})$. □

3.4. LEMMA. *The number of ones in the first half of the permutation P_k is $\Omega(N)$.*

Proof. We partition the permutation into lines (as above) and calculate the number of elements in the first 2^{k-4} lines that are expressible as a linear combination (using non-negative integer coefficients) of h_k, h_{k+1}, and h_{k+2}. This number is clearly a lower bound for the total number we need to show to establish the lemma. As in the proof of Lemma 3.3, we have the three equations,

$$\begin{aligned} p + 4q + 16r &= \alpha \\ p + 2q + 4r &= \beta \\ p + q + r &= \gamma, \end{aligned}$$

with $\alpha \le 2^{k-4}$ and we need to lower-bound $|(\alpha, \beta, \gamma)|$. Each triple (p, q, r) generates a unique triple (α, β, γ), since the three equations above are independent. Thus we only need to derive a lower bound for the number of (*integral*) triples (p, q, r). The equation

$$p + 4q = L$$

clearly has about $L/4$ solutions, so for each $0 \le r \le \alpha/16$, there are about $(\alpha - 16r)/4$ solutions. Thus (with Θ notation implied), for each line α:

$$\begin{aligned} |(\beta, \gamma)| &= \sum_{r=0}^{\alpha/16} \frac{\alpha - 16r}{4} \\ &= \frac{\alpha^2}{64} - \sum_{r=0}^{\alpha/16} 4r \\ &= \frac{\alpha^2}{64} - \frac{\alpha^2}{128} \\ &= \frac{\alpha^2}{128}. \end{aligned}$$

Thus each line $\alpha \leq 2^{k-4}$ has about $\alpha^2/128$ ones. Thus there are

$$\sum_{\alpha=0}^{2^{k-4}} \frac{\alpha^2}{128} = \frac{8^{k-4}}{384} = \Omega(N)$$

ones in the first 2^{k-4} lines, proving the lower bound and hence the lemma. □

It is now easy to prove that this permutation has a quadratic number of inversions.

3.5. LEMMA. *The number of inversions in permutation P_k is $\Omega(N^2)$.*

Proof. By Lemma 3.4, there are $\Omega(N)$ ones in the first half of P, which implies $\Omega(N)$ zeros in the second half. (See [9] for a quick proof of this.) Thus there are $\Omega(N^2)$ inversions. □

Remark. The constant implied in this proof is quite small, because only the ends of P_k are considered. Empirical evidence strongly suggests that the number of inversions tends to $N^2/48$. Proving this would require a much tighter argument than the one above.

We are now ready to prove the first main result of this paper.

3.6. THEOREM. *The running time for Shellsort is $\Omega(N^{4/3})$ for the increments $1, 8, 23, 77, \ldots, h_k = 4^{k-1} + 3 \cdot 2^{k-2} + 1, \ldots$.*

Proof. If Shellsort is run on P_k, no exchanges are performed during the h_t-*sort*, h_{t-1}-*sort*, $\ldots$, h_k-*sort*, and hence no inversions are removed during these passes. It follows, from the swapping lemma, that at most h_{k-1} inversions can be removed during any exchange. Thus the number of exchanges necessary is $\Omega(N^2/h_{k-1})$. We know that $h_{k-1} = \Theta(N^{2/3})$, hence we obtain the lower bound of $\Omega(N^{4/3})$, completing the proof. □

Remark 1. If we want to prove this result for a general permutation of N integers, we proceed as follows: Assign the largest integers to the ones, and the smallest integers to the zeros. The particular order is unimportant. When we come to h_{k-1}-*sort*, we still have a quadratic number of inversions, and we can remove them only twice as fast as before. Hence the bound still holds.

Remark 2. For any arbitrary N we can obtain the lower bound merely by padding a smaller Frobenius pattern with ones, affecting only the constant in the bound. However, we can do better. Take the next highest N' that is a Frobenius number of $h_j h_{j+1}, \ldots, h_\infty$ and use the middle N elements of P_j. This sequence will have $\Omega(N^2)$ inversions. To see this, note that an extension of Lemma 3.4 is certainly true, because the number

of ones on a line can only increase as the line number gets bigger. Thus, there is a greater density of ones near the middle of the permutation than at the ends. Lemma 3.4 implies Lemma 3.5. Moreover, this permutation will also satisfy Lemma 4, so we obtain an $\Omega(N^{4/3})$ lower bound for any N.

4. More Lower Bounds

The general technique used in the previous section can be extended to prove lower bounds for other increment sequences.

For instance, the increment sequence $1, 65, \ldots, h_k = (2^k - 3)(2^{k+1} - 3)(2^{k+2} - 3) = 8 \cdot 8^k - 42 \cdot 4^k + 63 \cdot 2^k - 27, \ldots$ can be shown to make Shellsort run in $\Omega(N^{5/4})$ in exactly the same manner as above [14, 17]. The increment sequences of Incerpi and Sedgewick that yield $O(N^{1+\varepsilon})$ upper bounds can likewise be proven tight. This is not surprising, since the general form of the upper bound given at the end of Section 2 and the general form of the lower bound are identical.

In general, suppose the increments h_k satisfy $h_k = \Theta(\alpha^k)$ for any (not necessarily integer) α. Suppose that h_t is the largest increment and that we use the permutation P_k as before. In this case, we need to obtain the maximum value of h_k to use in generating P_k. $g(h_k, h_{k+1}, \ldots, h_t) = N - 1$ and thus the lower bound of Incerpi and Sedgewick implies that

$$h_k^{1+1/(t-k)} = O(N).$$

On the other hand, it is also true that

$$h_k = \Theta\left(\frac{N}{\alpha^{t-k}}\right).$$

Combining these equations, we obtain

$$t - k = \Omega\left(\sqrt{\log_\alpha N}\right)$$

which yields

$$h_k = O\left(N^{1-1/\sqrt{\log_\alpha N}}\right).$$

Thus, if P_k has $\Omega(N^2)$ inversions, we obtain a lower bound of $\Omega(N^{1+1/\sqrt{\log_\alpha N}}) = \Omega(N^{1+\varepsilon/\sqrt{\log N}})$ which matches the best known upper bound for $O(\log N)$ increment sequences. We cannot prove that P has $\Omega(N^2)$ inversions, but we make the following conjecture which would be

sufficient to prove this result:

4.1. *Inversion Conjecture.* Given $a_1 < a_2 < \cdots < a_k$, then the number of inversions in the Frobenius pattern (of size N) formed from these integers is $\Theta(N^2/k)$.

For $k = 2$, the conjecture is easily proven. For other values of k, empirical evidence strongly suggests that the conjecture is true; in fact, the implied constant seems to be $\frac{1}{24}$. Moreover, to obtain the lower bound for Shellsort, all we need is the following weak form of the inversion conjecture which must certainly be true:

4.2. *Weak Inversion Conjecture.* Given $a_1 < a_2 < \cdots < a_k$ then the number of inversions in the Frobenius pattern formed from the integers is $\Omega(N^2/f(k))$, with $f(k) = o(2^k)$.

We then have the following theorem:

4.3. THEOREM. *The running time for Shellsort is* $\Omega(N^{1+\varepsilon/\sqrt{\log N}})$ *for increments* $h_k = \Theta(\alpha^k)$ *for any* $\alpha > 1$ *if the* weak inversion conjecture *is true.*

Proof. By the discussion above, if the number of inversions is $\Omega(N^2/f(k))$, we obtain a running time of $\Omega(N^{1+\varepsilon/\sqrt{\log N}}/f(\sqrt{\log_\alpha N}))$. If $f(k) = o(2^k)$, this is still $\Omega(N^{1+\varepsilon'/\sqrt{\log N}})$ for some $0 < \varepsilon' < \varepsilon$. □

Remark. If the number of inversions is $\Theta(N^2/2^k)$, then we obtain the trivial lower bound of $\Omega(N)$ because $2^k = 2^{\sqrt{\log_\alpha N}} = N^{1/\sqrt{\log_\alpha N}}$.

5. CONCLUSIONS AND OPEN PROBLEMS

We have shown tight lower bounds for Shellsort using a wide range of increment sequences. A similar technique can be used to show that Shaker-sort, which is a network sorter based on Shellsort (probably) has a quadratic worst case when the increments are almost geometric [19]. The proof of this claim depends on the (weak) inversion conjecture.

Some interesting open problems remain. First and foremost is proving our inversion conjecture, or any somewhat weaker form as suggested in Section 4. Assuming the inversion conjecture, proving that even if only some subset of increments is $\Theta(\alpha^k)$, then Shellsort is $\Omega(N^{1+\varepsilon/\sqrt{\log N}})$ would generalize our result quite a bit. This would take care of some $O(\log N)$ increment sequences that do not strictly increase. It turns out that for many of these increment sequences, we can still prove the lower bound but we need a slightly different proof; a unifying concept would be nice.

ACKNOWLEDGMENTS

We thank John Comfort and an anonymous referee for their many constructive comments.

REFERENCES

1. A. BRAUER, On a problem of partitions, *Amer. J. Math.* **64** (1942), 299–312.
2. R. CYPHER, A lower bound on the size of Shellsort sorting networks, *in* "Proceedings, 1989 ACM Symp. on Parallel Algorithms and Architectures, Santa Fe, NM, June 1989."
3. G. GONNET, "Handbook of Algorithms and Data Structures," Addison–Wesley, Reading, MA, 1984.
4. T. N. HIBBARD, An empirical study of minimal storage sorting, *Comm. ACM* **6**, No. 5 (1963), 206–213.
5. J. INCERPI, "A Study of the Worst-Case of Shellsort," Ph.D. thesis, Brown University, 1985.
6. J. INCERPI AND R. SEDGEWICK, Improved upper bounds on Shellsort, *J. Comput. System Sci.* **31**, No. 2 (1985), 210–224.
7. S. M. JOHNSON, A linear diophantine problem, *Canad. J. Math.* **12** (1960), 390–398.
8. D. E. KNUTH, "The Art of Computer Programming, Vol. 3. Sorting and Searching," Addison–Wesley, Reading, MA, 1973.
9. A. NIJENHUIS AND H. S. WILF, Representations of integers by linear forms in nonnegative integers, *J. Number Theory* **4** (1972), 98–106.
10. A. A. PAPERNOV AND G. V. STASEVICH, A method of information sorting in computer memories, *Problems Inform. Transmission* **1**, No. 3 (1965), 63–75.
11. V. PRATT, "Shellsort and Sorting Networks," Garland, New York, 1979; Ph.D. thesis, Stanford University, 1971.
12. R. SEDGEWICK, A new upper bound for Shellsort, *J. Algorithms* **2** (1986), 159–173.
13. E. S. SELMER, On the linear diophantine problem of Frobenius, *J. Reine Angew. Math.* **294** (1977), 1–17.
14. E. S. SELMER, On Shellsort and the Frobenius problem, *BIT* **29**, No. 1 (1989), 37–40.
15. W. J. CURRAN SHARP, Solution to Problem 7382 (Mathematics), *Ed. Times, London* (1884).
16. D. L. SHELL, A high-speed sorting procedure, *Comm. ACM* **2**, No. 7 (1959), 30–32.
17. M. A. WEISS AND R. SEDGEWICK, More on Shellsort increment sequences, *Inform. Process. Lett.*, to appear.
18. M. A. WEISS, A good case for Shellsort, *Congressus Numerantium* **73** (1989), 59–62.
19. M. A. WEISS AND R. SEDGEWICK, Bad cases for Shaker sort, *Inform. Process. Lett.* **18** (1988), 133–136.