Information Processing Letters 34 (1990) 267–270
North-Holland

7 May 1990

MORE ON SHELLSORT INCREMENT SEQUENCES

Mark Allen WEISS *

School of Computer Science, Florida International University, University park, Miami, FL 33199, USA

Robert SEDGEWICK **

Department of Computer Science, Princeton University, Princeton, NJ 08540, USA

Communicated by D. Gries
Received 19 June 1989

Keywords: Shellsort, sorting

1. Introduction

Shellsort is a sorting algorithm proposed by Shell [9] in 1959. Using a sequence of integers $h_1, h_2, \ldots, h_t$, Shellsort works by performing insertion sort on subfiles consisting of elements h_i apart. We call this an h_i-*sort*. Shellsort works by performing an h_t-*sort*, an h_{t-1}-*sort*, and so on until an $h_1 = 1$-*sort*. Typically the increment sequence is "almost" geometric, so there are $O(\log N)$ increments (or passes).

For some time, the best known bound for the worst-case running time of Shellsort with $O(\log N)$ increments was due to Pratt [6] who showed that if $h_i = c \cdot \alpha^i + d$ for some integer α (i.e., the increments were within an additive constant of geometric), then the running time of Shellsort is bounded by $\Theta(N^{3/2})$. Sedgewick [7] provided two sets of $O(\log N)$ increments each with worst-case bound $O(N^{4/3})$; this bound was proven tight by Weiss and Sedgewick [10].

Incerpi and Sedgewick [4] subsequently (although their paper appeared earlier) provided two families of increment sequences, and proved an $O(N^{1+1/(c+1)})$ bound for the first family and an $O(N^{1+\epsilon/\sqrt{\log N}})$ bound for the second (a third nonuniform increment sequence was presented, but we will not deal with it here). In proving these bounds, Incerpi and Sedgewick developed a "generalized Frobenius function", thus circumventing the usual Frobenius function used for all previous proofs.

Section 2 reviews previous results on the Frobenius problem and their use in bounding Shellsort's running time. Section 3 presents simpler proofs of the bounds obtained by Incerpi and Sedgewick. Our proofs use Johnson's classic result on the Frobenius problem. Although our results do not improve the bounds, they extend the bound to cover more increment sequences and show that the bounds for all known increment sequences can be proven using a standard Frobenius argument. There are many sequences that satisfy the conditions required in the proof and only one was actually tried in the original paper. Thus, there was a real possibility that a sequence of this form existed that would give a major improvement in the running time of Shellsort in practice. We have tried the possibilities of interest and Section 4 details the results of our empirical studies. Section 5 offers conclusions and open problems.

2. Previous bounds

Central to proving the upper bounds in the Frobenius number defined as follows:

* Supported by an FIU Foundation Summer Research Grant.

** Supported by National Science Foundation Grant DCR-8605962.

Definition 2.1. $g(a_1, a_2, \ldots, a_k) \equiv$ the largest integer which cannot be represented as a linear combination, with nonnegative integer coefficients, of $a_1, a_2, \ldots, a_k$.

The Frobenius number is defined if and only if $gcd(a_1, a_2, \ldots, a_k) = 1$. We will need two well-known results on the Frobenius problem:

Lemma 2.2 (Curran-Sharp, 1884). *If* $gcd(a_1, a_2) = 1$, *then* $g(a_1, a_2) = (a_1 - 1)(a_2 - 1) - 1$.

Proof. See [2]. □

Lemma 2.3 (Johnson, 1960). *If g is defined and* $d = gcd(a_1, a_2, \ldots, a_{k-1})$, *then*

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

Proof. See [5]. □

One way to prove the upper bounds for Shellsort, is to determine (or at least bound) the function

$$N \frac{g(h_{k+1}, h_{k+2}, \ldots, h_t)}{h_k} \tag{2.1}$$

for each increment h_k. By using this bound for "small" increments and the obvious $O(N^2/h_k)$ bound for "large" increments, the running time is bounded (apparently tightly) [10].

Incerpi and Sedgewick defined the "generalized Frobenius function" as follows:

Definition 2.4. $n_d(a_1, a_2, \ldots, a_k) \equiv$ the number of multiples of d which cannot be represented as a linear combination (with nonnegative integer coefficients) of $a_1, a_2, \ldots, a_k$.

Using the generalized Frobenius function, they replaced (2.1) with

$$N \cdot n_{h_k}(h_{k+1}, h_{k+2}, \ldots, h_t). \tag{2.2}$$

We will show that for the uniform increments in [4], we obtain the same value for (2.1) that was obtained by Incerpi and Sedgewick for (2.2), thereby proving their bounds in the more standard manner. These results are rather surprising since although the "generalized" Frobenius function seems to be a much more accurate bound we are not only able to obtain identical bounds, but are able to obtain them for a far greater range of sequences.

3. Alternative proofs

In this section we prove upper bounds for the uniform increment sequences suggested by Incerpi and Sedgewick. Our two proofs are similar in that they are by induction and use Johnson's formula for removing common divisors.

Theorem 3.1. *Let* $a_1, a_2, \ldots$ *be a sequence of pairwise relatively prime integers* (*with* $a_1 = 1$), *and let c be a fixed constant. Let* $h_k = \prod_{j=k-1}^{k+c-2} a_j$ *for* $k > 2$. *Then*

$$g(h_{k+1}, h_{k+2}, \ldots, h_{k+c}) = \Theta\left(\prod_{j=k+c-1}^{k+2c-1} a_j\right).$$

Proof (By induction). The basis $c = 1$ is trivial. Assume the theorem is true for $c - 1$. Then Johnson's formula applies because $h_{k+2}, h_{k+3}, \ldots, h_{k+c+1}$ all share a common factor, a_{k+c}, that h_{k+1} doesn't and furthermore, g is defined because its arguments are relatively prime. Thus,

$$g(h_{k+1}, h_{k+2}, \ldots, h_{k+c+1}) = a_{k+c} g\left(h_{k+1}, \frac{h_{k+2}}{a_{k+c}}, \frac{h_{k+3}}{a_{k+c}}, \ldots, \frac{h_{k+c+1}}{a_{k+c}}\right) + (a_{k+c} - 1)h_{k+1}.$$

Since $h_{k+1} = a_k(h_{k+2}/a_{k+c})$, h_{k+1} can be removed. Furthermore

$$\frac{h_{k+i}}{a_{k+c}} = \prod_{\substack{j=k+i-1 \\ j \neq k+c}}^{k+i+c-2} a_j$$

and thus by the induction hypothesis,

$$g\left(\frac{h_{k+2}}{a_{k+c}}, \frac{h_{k+3}}{a_{k+c}}, \ldots, \frac{h_{k+c+1}}{a_{k+c}}\right) = \Theta\left(\prod_{\substack{j=k+c-1 \\ j \neq k+c}}^{k+2c-1} a_j\right).$$

The result

$$g(h_{k+1}, h_{k+2}, \ldots, h_{k+c+1}) = \Theta\left(\prod_{j=k+c-1}^{k+2c-1} a_j\right)$$

is immediate. □

If follows that if $a_1, a_2, \ldots, a_k$ are all within a constant factor of each other, then equation (2.1) evaluates to $O(N \cdot h_k^{1/c})$. This result is used to bound Shellsort's running time. As in the original proof, the result for equation (2.1) and hence the final Shellsort bound hides a constant that is exponential in c.

Incerpi and Sedgewick's sequence was a merge of sequences like the one above and thus their sequence and proof was unnecessarily complicated. This new proof provides an enormous set of increment sequences that are possibly better on average than any known. For example, one might try increment sequences of the form $h_i = a_k c^{ik} + a_{k-1} c^{i(k-1)} + \cdots + a_0$. Selmer [8] has independently obtained this result by using a formula due to Brauer [1].

Theorem 3.2. *Let $a_1, a_2, \ldots$ be a sequence of pairwise relatively prime integers (with $a_1 = 1$), let $c > 2$ be an integer and let $x_i = \prod_{j=1, j \neq c-i+1}^{c} a_j$. Then*

$$g(x_1, x_2, \ldots, x_{c-1}) \leqslant (c-1) \prod_{j=1}^{c} a_j.$$

Proof (by induction). The basic $c = 3$ is true because $g(a_1a_2, a_1a_3) = g(a_2, a_3) < a_1a_2a_3$. Assume the theorem is true for $c-1$. $gcd(x_2, x_3, \ldots, x_{c-1}) = a_c$ and g is defined so Johnson's formula can be applied, yielding

$$g(x_1, x_2, \ldots, x_c) = a_c g\left(x_1, \frac{x_2}{a_c}, \frac{x_3}{a_c}, \ldots, \frac{x_c}{a_c}\right) + (a_c - 1)x_1.$$

Proceeding as in Theorem 3.1, we can use the inductive hypothesis to replace the Frobenius function on the left with $(c-2)\prod_{i=1}^{c-1} a_i$, and the required result follows immediately. □

For any increment h_c which is a product of the first c terms of the base sequence except for one term, a_p, bound (2.1) becomes $O(cN \cdot a_p a_{c+1})$. If the base sequence has $a_i = O(\alpha^i)$, as in [4], then for any $\epsilon > 0$ we can choose $\alpha < 2^{\epsilon^2/8}$ and obtain the $O(N^{1+\epsilon/\sqrt{\log N}})$ bound. Note that our bound (2.1) has an extra factor of c that is not found in bound (2.2). We have thus chosen $\alpha < 2^{\epsilon^2/8}$ instead of the original choice $\alpha = 2^{\epsilon^2/8}$.

The third increment sequence, due to Chazelle, can also be bounded using this technique. We don't include it here because our proof seems more complicated than the original.

4. Practical applications

Unspecified in the proofs above is the actual base increment sequence $a_1, a_2, \ldots$. The only requirement is that this sequence is increasing at most geometrically and has terms that are pairwise relatively prime. For the first set of increments, there is no good choice for this sequence because the first few increments are too large. For instance, if $c = 4$ and $a_1 = 2$, $a_2 = 3$, $a_3 = 5$ and $a_5 = 7$, then the first increment is 210. It is still possible however to adjust some of the first few increments to make the algorithm better in practice, but the upper bounds might not hold.

For the second set of increments, Incerpi and Sedgewick took a_{i+1} = smallest prime $\geqslant 2^i$ = {1, 2, 5, 11, ...} obtaining sorts on 20000 elements that required 489000 comparisons on average. (Comparisons are directly proportional to actual running time.) We have found that better choices for the base sequence are possible, leading to slight improvements in actual performance. If we are interested in file sizes for which Shellsort is practical, we need consider only the first few terms of the base sequence. For sorting 20000 elements, Incerpi and Sedgewick needed only the first six terms of the base sequence; only {1, 2, 5, 11, 17, 37} are used. The most important factor seems to be avoiding $a_2 = 2$. Both a_{i+1} = smallest prime $> \alpha^i$ with $2 < \alpha < 3$ and a_{i+1} = largest prime $< \alpha^{i+1}$ with $\sqrt{3} < \alpha \leqslant 2$ satisfy this property. It turns out to be easy (computationally)

to generate all the sequences defined above and run Shellsort using each implied increment sequence on 20000 elements because we are rounding up (or down) to a prime and many different values of α generate the same base sequence (for example $\alpha = 2.003$ and $\alpha = 2.004$). For $53^{1/6} < \alpha < 29^{1/5}$, using the second alternative, a base sequence of $\{1, 3, 7, 13, 23, 53, \ldots\}$ is obtained requiring 465255 comparisons on average. This represents a 5% improvement over the original choice. Similar results (within $\frac{1}{2}$%) are obtained if the base sequence starts with $\{1, 3, 7, 13\}$, regardless of what the next term is.

Another possibility is to exhaustively generate all base sequences with five non-one terms all pairwise relatively prime and determine the best choice as above. To make this calculation run in reasonable time (8 days), we eliminated certain base sequences that had little chance of being good even though the relative primeness condition was satisfied (for instance, the sixth term of the base sequence had to be at least 1.4 times as large as the fifth term). It is unlikely that this assumption caused the loss of a good sequence because all of the sequences on the "fringe" were themselves poor. This method yields an additional 1% improvement for several choices. One such choice is $\{1, 4, 9, 17, 23\ldots\}$ which required 460948 comparisons on average for the 200 random permutations tested. Choosing 4 as the second term in the base sequence seems to be best in general.

A third possibility is to not insist that the relative primeness condition always hold (sacrificing the theoretical upper bound), but it turns out that this does not work well because eventually all sufficiently large terms will share a common factor.

5. Summary

We have shown that upper bounds using complicated increment sequences can be proven using the standard Frobenius function argument. We conjecture that all upper bounds can be derived in the same manner and that the upper bound for $O(\log N)$ increment sequences cannot be improved. Lower bound arguments supporting this conjecture can be found in [10].

We have also obtained slightly better performance for the Incerpi–Sedgewick increment sequences, but this must be considered a negative result in that none of the increment sequences represent a major improvement in the running time of Shellsort and are in fact slower than some previously known sequences [7]. It is not likely that these types of sequences can be made any better in practice, since we have tried almost all possibilities.

Acknowledgment

Erich Hentschel wrote the code and ran the tests described in Section 4, producing several hundred pages of output.

References

[1] A. Brauer, On a problem of partitions, *Amer. J. Math.* **64** (1942) 299–312.
[2] W.J. Curran Sharp, Solution to Problem 7382 (Mathematics), *Educational Times*, London (1884).
[3] J. Incerpi, A study of the worst-case of shellsort, Ph. D. Thesis, Brown University, Providence, RI (1985).
[4] J. Incerpi and R. Sedgewick, Improved upper bounds on Shellsort, *J. Comput. System Sci.* **31** (2) (1985) 210–224.
[5] S.M. Johnson, A linear diophantine problem, *Canad. J. Math.* **12** (1960) 390–398.
[6] V. Pratt, *Shellsort and Sorting Networks* (Garland, New York, 1979).
[7] R. Sedgewick, A new upper bound for Shellsort, *J. Algorithms* **2** (1986) 159–173.
[8] E.S. Selmer, On Shellsort and the Frobenius problem, *BIT* **1** (1989) 37–41.
[9] D.L. Shell, A high-speed sorting procedure, *Comm. ACM* **2** (7) (1959) 30–32.
[10] M.A. Weiss, Tight lower bounds for Shellsort, *J. Algorithms,* to appear.