Comparison of $\Theta(n \lg n)$ sortings

For in-class use only in CSC 501/401 course

Dr. Marek A. Suchenek ©

November 24, 2018

Abstract

All $\Theta(n \lg n)$ sorting algorithms discussed in class perform less than $2n \lg n$ comparisons of keys in order to sort any n-element array of distinct elements.

Some of those perform slightly less than $n \lg n$ comparisons of keys.

The state of the art in the question of optimality of sorting by decision tree is summarized in this paper.

1 Quicksort

The worst-case number of comparisons of keys performed by basic Quicksort while sorting an array of n distinct elements is rather disappointing and equal to:

$$\frac{n(n-1)}{2}.$$

The average-case number is equal to

$$2(n+1)\sum_{i=1}^{n}\frac{1}{i} - 4n \approx 1,386(n+1)\lg n - 2.846n + 2.154 + \frac{1}{n}.$$

The best-case number is equal to

$$\sum_{i=1}^{n}\lfloor \lg i \rfloor = (n+1)(\lg(n+1)+\varepsilon(n+1))-2n = (n+1)(\lg\frac{n+1}{4}+\varepsilon(n+1))+2,$$

where ε, given by:

$$\varepsilon(n) = 1+\theta-2^{\theta} \text{ and } \theta = \lceil \lg\ n \rceil - \lg\ n,$$

is a continuous function of n on the set of reals > 1, with the minimum value 0 and and the maximum (*supremum*, if n is restricted to integers) value

$$\delta = 1 - \lg e + \lg\lg e \approx 0.0860713320559342.$$

2 Mergesort

The worst-case number of comparisons of keys performed by Mergesort while sorting an array of n distinct elements is equal to:

$$\sum_{i=1}^{n} \lceil \lg i \rceil = n(\lg n + \varepsilon(n)) - n + 1 = n(\lg \frac{n}{2} + \varepsilon(n)) + 1,$$

where ε, given by:

$$\varepsilon(n) = 1 + \theta - 2^{\theta} \text{ and } \theta = \lceil \lg\ n \rceil - \lg\ n,$$

is a continuous function of n on the set of reals > 1, with the minimum value 0 and and the maximum (*supremum*, if n is restricted to integers) value

$$\delta = 1 - \lg e + \lg \lg e \approx 0.0860713320559342.$$

The best-case number is given by a rather convoluted formula (to appear in my forthcoming text) and is slightly less than $\frac{1}{2} n \lg n$ as the Fig. 1 below shows.

3 Heapsort

The worst-case number of comparisons of keys performed by basic Heapsort while sorting an array of n distinct elements is given by a formula that I

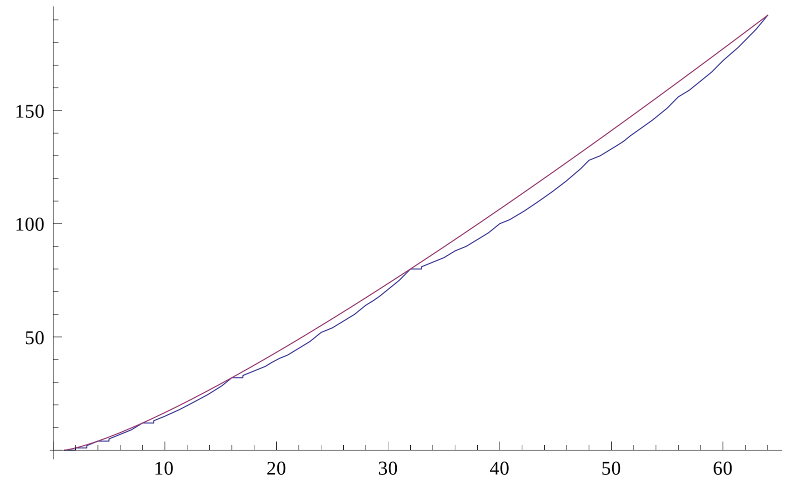

Figure 1: The best-case number of comparisons for Mergesort plotted below function $\frac{1}{2}n \lg n$.

discovered in Summer of 2013 (to be published in my forthcoming paper). The formula is a bit complicated as its graph on Fig. 2 below suggests.

Its fairly good upper bound is given by

$$2n \lg n$$

as it is shown on Fig. 3 below.

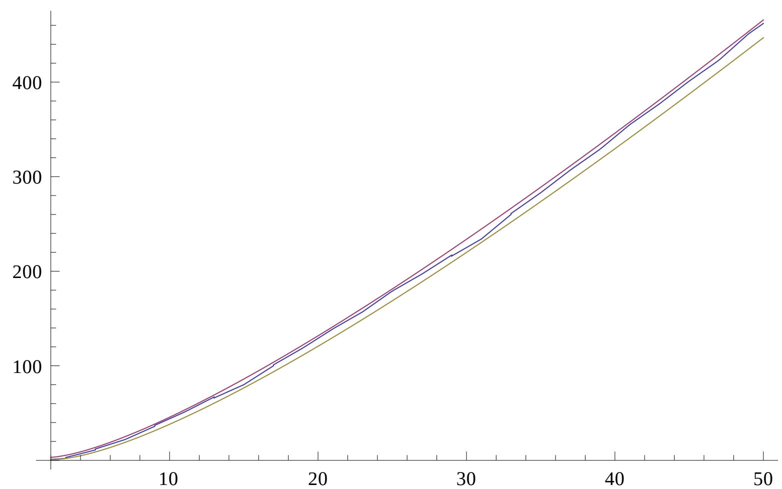

Figure 2: The worst-case number of comparisons of `HeapSort` plotted against its upper $2(n-1)(\lg(n-1) - 0.9139)$ and lower $2(n-1)(\lg(n-1) - 1) - 2\lg(n+1) + 6$ bounds.

4 Accelerated Heapsort

Accelerated Heapsort uses slightly more than half of the number of comparisons of keys that the basic Heapsort (with fast MakeHeap) performs in the worst case. According to your textbook [BG00], the number of comparisons

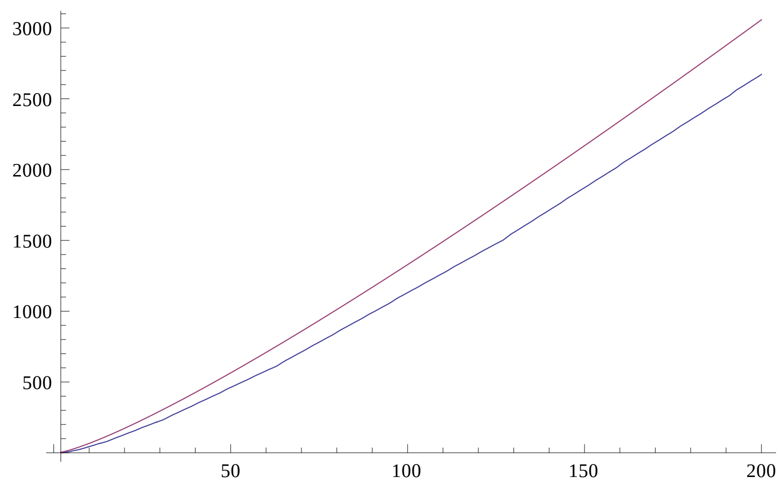

Figure 3: The worst-case number of comparisons of `HeapSort` plotted against its upper bound $2n \lg n$.

done by accelerated Heapsort is no more than

$$n \lg n + O(n \lg \lg n).$$

Any improvements to the aforementioned upper bound will be posted here as they become available.

5 The least known upper bound for the worst case

The worst-case provably fastest[1] known uniform algorithm[2] that sorts by decision tree is based on Merge-insertion sort (not covered in class). Merge-insertion sort performs

$$\sum_{i=1}^{n} \lceil \lg \frac{3}{4} i \rceil$$

comparisons of keys in the worst case while sorting an array of n distinct elements. For years it was believed to be the worst-case optimal[3] sorting algorithm, which conjecture has been proven false in 1979 (in general) and 1981 (for $n = 47$). Interestingly enough, in the latter case, Merge-insertion sort lost to itself (by just one comparison) in a sense that sorting 5 and 42 elements using it and then merging the results efficiently required only 200 comparisons in the worst case while Merge-insertion sorting of 47 elements required 201. This leaves the following as the least known upper bound $F(n)$ for the worst case:

$$F(n) = \begin{cases} \sum_{i=1}^{n} \lceil \lg \frac{3}{4} i \rceil \text{ if } n \neq 47 \\ \\ 200 \text{ if } n = 47. \end{cases}$$

[1] In terms of the number of comparisons of keys performed in the worst case.
[2] An algorithm that does not distinguish special cases of n.
[3] In class of algorithms that sort by decision tree

Figures 4 and 5 show graphs of the upper bound $F(n)$ plotted against the information-theoretic lower bound.

It is known since 1979 that for infinitely many sizes of sorted arrays, Merge-insertion sort is not worst-case optimal.

6 Lower bounds

The information-theoretic lower bound on the worst-case number of comparisons of keys performed by any sorting algorithm that sorts by decision tree while sorting an array of n distinct elements is equal to:

$$\lceil \lg n! \rceil > \lceil (n + \frac{1}{2}) \lg n - 1.443n + 1.325 \rceil.$$

The information-theoretic lower bound on the average-case number of comparisons of keys performed by any sorting algorithm that sorts by decision tree while sorting an array of n distinct elements is equal to:

$$\lg n! + \varepsilon(n!) > (n + \frac{1}{2}) \lg n - 1.443n + 1.325.$$

where ε, given by:

$$\varepsilon(n) = 1 + \theta - 2^{\theta} \text{ and } \theta = \lceil \lg\ n \rceil - \lg\ n,$$

is a continuous function of n on the set of reals > 1, with the minimum value 0 and and the maximum (*supremum*, if n is restricted to integers) value

$$\delta = 1 - \lg e + \lg \lg e \approx 0.0860713320559342.$$

The information-theoretic lower bound on the best-case number of comparisons of keys performed by any sorting algorithm that sorts by decision tree while sorting an array of n distinct elements is bit tricky to define. On one hand, it is obvious that at least $n-1$ comparisons are required to verify that the array in question is sorted. On the other hand, an algorithm that performs $n-1$ comparisons in the best case cannot be worst-case or average-case optimal for all n; for instance, such an algorithm is not worst-case optimal for $n = 4, ..., 11, 20, 21$ and not average-case optimal for $n = 4, 5, 9, 10$.

It does make sense to define the lower bound on the best-case number of comparisons as the minimal number of comparisons that an average-case (alternatively: a worst-case) optimal algorithm that sorts by decision tree must perform while sorting an n-element array.

If the algorithm in question is information-theoretic optimal on the average then the following is the number of comparisons that the said algorithm must perform in the bast case[4]:

$$\lfloor \lg((n! - 1) + 1) \rfloor = \lfloor \lg n! \rfloor > \lfloor (n + \frac{1}{2}) \lg n - 1.443n + 1.325 \rfloor,$$

which for $n > 2$ is exactly one less than in the worst case[5].

[4]The number of comparisons that the said algorithm must perform in the bast case is equal to $\lfloor \lg(k + 1) \rfloor$, where k is the number of internal nodes in its decision tree, or $k = n! - 1$.

[5]Because for $n > 2$, $n!$ is never a power of 2

If the algorithm in question is information-theoretic optimal in the worst case then the number of comparisons $c(n)$ that the said algorithm must perform in the bast case oscillates between $n-1$ and $\lfloor \lg n! \rfloor$ as n diverges to ∞:

$$n - 1 \leq c(n) \leq \lfloor \lg n! \rfloor.$$

Although for some specific sizes (e.g., 1 through 11, 20, 21 for the worst-case lower bound and 1 through 5, 9, 10 for the average-case lower bound) of the sorted array the information-theoretic lower bounds can be actually reached by sorting algorithms that sort by decision tree, not much more is known in this respect (consult [Knu97] for more details).

Figures 4 and 5 visualize the numbers of comparisons $\sum_{i=1}^{n} \lceil \lg \frac{3}{4} i \rceil$ performed by Merge-insertion sort (the worst-case-fastest known sorting algorithm that sorts by decision table) in the worst case (the upper dots) plotted above the information-theoretic lower bound $\lceil \lg n! \rceil$ (the lower dots). They illustrate the current knowledge in this respect. When both dots in a vertical line on the graphs plotted on these Figures coincide then the respective information-theoretic lower bound is reached by Merge-insertion sort. Otherwise, it is not. In the latter case, ! indicates the optimum[6] and ? indicates the suspected optimum. If there is no indication then any of the two dots (it is not known which one, though) or a value between them may represent the optimum.

[6]Except for $n = 47$ where ! indicates the information-theoretic lower bound of 198 that is not reached by Merge-insertion sort.

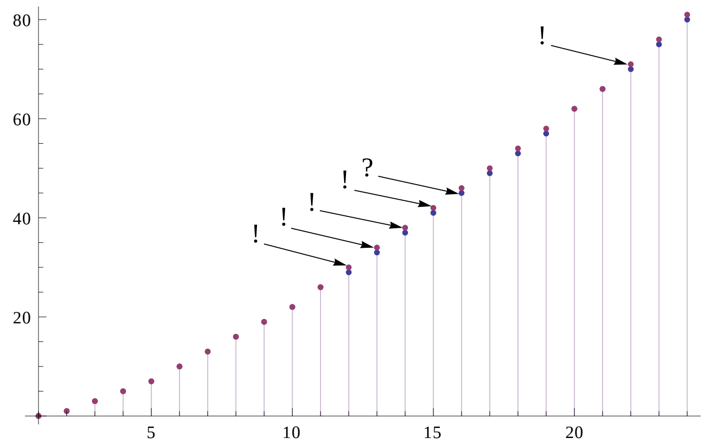

Figure 4: Number of comparisons performed by Merge-insertion sort in the worst case (upper dots) plotted above the information-theoretic lower bound (lower dots). ! indicates the actual optimum. ? indicates suspected optimum.

For instance, it is not known what is the actual minimum number of comparisons that suffices to sort any array of 16 distinct elements[7]. The information-theoretic minimum is 45 comparisons while the fastest known sorting of 16 distinct elements takes 46 comparisons. Knuth in [Knu97], p. 192, conjectures that the said minimum is 45. A recent paper by Marcin

[7] $16! = 20\,922\,789\,888\,000$.

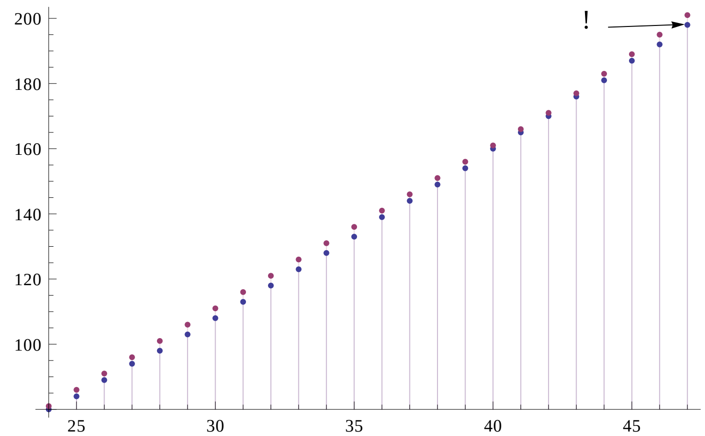

Figure 5: Number of comparisons performed by Merge-insertion sort in the worst case (upper dots) plotted above the information-theoretic lower bound (lower dots) - continued from Fig. 4. ! indicates the lower bound at $n = 47$ for which Merge-insertion sort is known to be not optimal.

Peczarski [Pec04] settled the question of worst-case optimality of sorting by decision table for $n = 13, 14$, and 22. Here is an excerpt from Peczarski's paper:

The reachability of the average-case optimality is even less known than of the worst-case. For instance, it is not know what is the minimum average number of comparisons that is needed to sort any array of 6 distinct an exhaustive computer search, that $S(12) = 30$. Knuth posed the problem of finding the next value $S(13)$ in his classic book [4, Chapter 5.3.1, Exercise 35]. He conjectured that $S(13) = 33$ and $S(14) = 37$ [6]. We show that $S(13) = 34$, $S(14) = 38$ and $S(22) = 71$. The Ford–Johnson algorithm turns out to be optimal for 13, 14 and 22

Figure 6: An excerpt from [Pec04] that disproves Knuth's conjectures $S(13) = 33$ (posed in [Knu97], p. 192, and disproved by Peczarski in his Master's Thesis in 2002) and $S(14) = 37$ (posed elsewhere).

elements[8]. The information-theoretic minimum is the average of 9.5777... comparisons while the fastest on the average known sorting of 6 distinct elements requires the average of 9.6 comparisons.

It is known, however, that the information-theoretic lower bounds cannot be reached for some specific sizes of sorted arrays. For instance, no algorithm that sorts by decision tree can perform less than 30 comparisons in the worst case while sorting an array of 12 distinct elements, while the information-theoretic lower bound for 12 elements is 29.

Also, any algorithm that sorts by decision tree must perform more than $\frac{3898}{315}$ comparisons on the average while sorting an array of 7 distinct elements, while the information-theoretic lower bound for 7 elements is $\frac{3898}{315}$.

The relationship between the worst-case and average-case optimality is

[8] $6! = 720$.

not clear as well.

What is known is that, for any size n of the sorted array, any algorithm that is information-theoretic optimal in the average case is also worst-case optimal, simply because any decision tree with $n!$ external nodes that has the shortest *external path length* is also a shortest decision tree with $n!$ external nodes.

However, it is not known if for all sizes n of the sorted array, average-case optimality for n implies worst-case optimality for n.

The converse implication does not hold, though, as some decision trees that are shortest may be not have the shortest *external path length*. An example in this category is Mergesort that for 6-element array is information-theoretic worst-case optimal but not average-case optimal.

7 Comparison of the results

Fig. 7 below shows graphs of numbers of comparisons of selected sorting algorithm and information-theoretic lower bounds for $n \leq 30$.

Fig. 8 shows the same graphs as Fig. 7 for $n \leq 30$, with added graphs of $2n \lg n$, $n \lg n$, and n for comparison of trends.

Fig. 9 shows the same graphs as Fig. 8 for $n \leq 300$.

Fig. 10 shows the same graphs as Fig. 9 for $n \leq 10,000$.

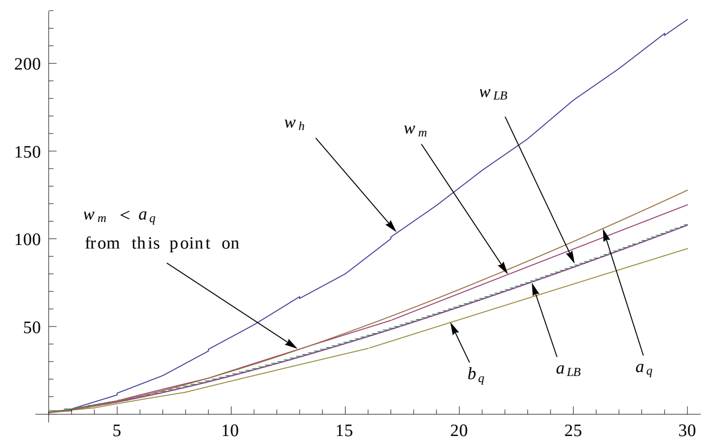

Figure 7: Number of comparisons performed by: Quicksort on average (a_q), Quicksort in the best case (b_q), Mergesort in the worst case (w_m), and Heapsort in the worst case (w_h), as well as the information-theoretic lower bounds on the worst-case (w_{LB}) and average (a_{LB}) number of comparisons.

.

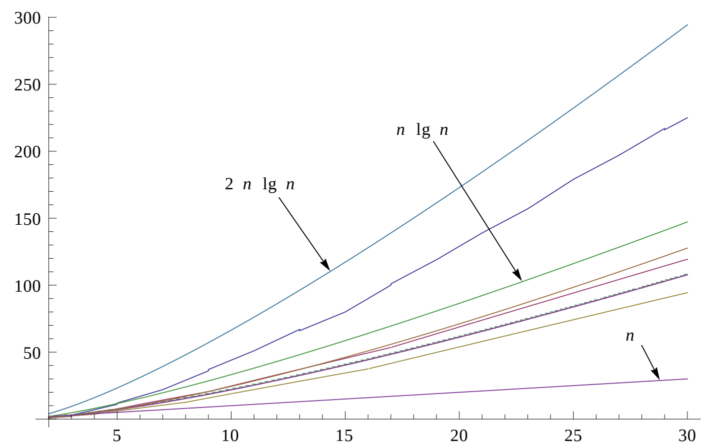

Figure 8: Number of comparisons performed by: Quicksort on average, Quicksort in the best case, Mergesort in the worst case, and Heapsort in the worst case, the information-theoretic lower bounds on the worst-case and average number of comparisons, and functions $2n \lg n$, $n \lg n$, and n.

.

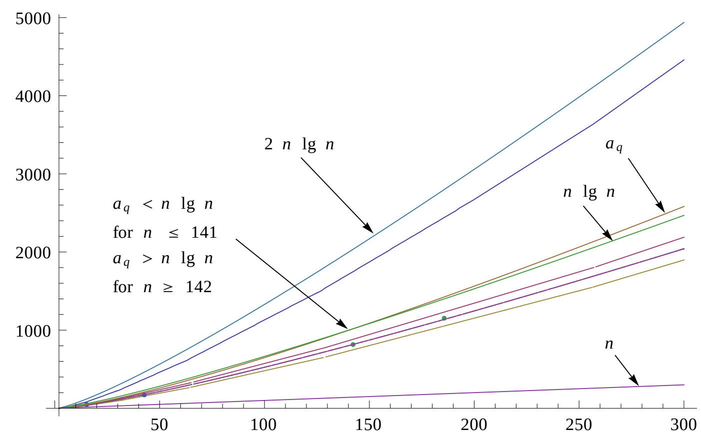

Figure 9: Number of comparisons performed by: Quicksort on average (a_q), Quicksort in the best case, Mergesort in the worst case, and Heapsort in the worst case, the information-theoretic lower bounds on the worst-case and average number of comparisons, and functions $2n \lg n$, $n \lg n$, and n.

.

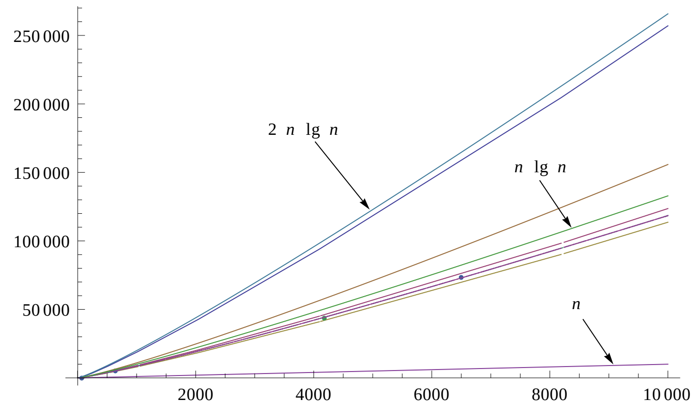

Figure 10: Number of comparisons performed by: Quicksort on average, Quicksort in the best case, Mergesort in the worst case, and Heapsort in the worst case, the information-theoretic lower bounds on the worst-case and average number of comparisons, and functions $2n \lg n$, $n \lg n$, and n.

References

[BG00] Sara Baase and Allen Van Gelder. *Computer Algorithms; Introduction to Design & Analysis.* Asddison Wesley, 3rd edition, 2000.

[Knu97] Donald E. Knuth. *The Art of Computer Programming*, volume 3. Addison-Wesley Publishing, 2nd edition, 1997.

[Pec04] Marcin Peczarski. New results in minimum-comparison sorting. *Algorithmica*, 40:133–145, 2004. DOI: 10.1007/s00453-004-1100-7.