On a Flaw in the Structure of Worst-case Heaps

MAREK A. SUCHENEK

California State University Dominguez Hills, Department of Computer Science, 1000 E. Victoria St., Carson, CA 90747, USA, Suchenek@csudh.edu

Abstract

A worst-case heap is defined as a max-heap $H$ of size $N$ that forces the sequence $H$.RemoveAll() of $N$ consecutive calls $H$.RemoveMax() to perform the maximum number of comps (comparisons of keys). Although it has been known that for $N \geq 3$, a single call $H$.RemoveMax() performs

$$C_{\text{RemoveMax}}(N) = \lfloor \lg(N-1) \rfloor + \lfloor \lg(N-2) \rfloor$$

comps in the worst case, it has also been known that $H$.RemoveAll() performs less than $\sum_{i=3}^{N} (\lfloor \lg(i-1) \rfloor + \lfloor \lg(i-2) \rfloor)$ comps if $N \geq 13$. Until recently, the exact number of comps performed by $H$.RemoveAll() on a worst-case heap of size $N$ had remain unknown, except when $N \leq 12$ or $N = 2^\lceil \lg N \rceil - 1$ (see [2] for an analysis of those special cases).

I am going to expose a singularity of worst-case heaps (discovered and proved in [4]). It states that if $N = 2^\lceil \lg N \rceil - 4$ and $H$ is any heap of size $N + 1$ such that $H$.RemoveMax() performs the worst-case number of comps (given by the equality (1)) then the heap produced by the call $H$.RemoveMax() is not a worst-case heap. It is a singular property, indeed, as for every $N \neq 2^\lceil \lg N \rceil - 4$, there is a worst-case heap $H$ of size $N + 1$ such that $H$.RemoveMax() on $H$ performs the worst-case number of comps and the heap produced by the call $H$.RemoveMax() is a worst-case heap.

The above allowed me to conclude (in [4]) that for every natural number $N \geq 2$, the number of comps performed by $H$.RemoveAll() on a worst-case heap $H$ of size $N$ is equal to:

$$2(N-1)[\lfloor \lg(N-1) \rfloor] - 2^\lceil \lg(N-1) \rceil + 2 + \min\{[\lg(N-1)], 2\} + 4 + c,$$

where $c$ is a binary function on the set of integers defined by:

$$c = \begin{cases} 
1 & \text{if } N \leq 2^\lceil \lg N \rceil - 4 \\
0 & \text{otherwise}. 
\end{cases}$$

Formula (2) together with a worst-case formula for MakeHeap (see [3] for its derivation) yield the following worst-case number of comps performed by Heapsort:

$$2(N-1)[\lg N] - 2^\lceil \lg N \rceil + 2s_2(N) - e_2(N) + \min\{[\lg N], 3\} + 5 + c,$$

Abstract submitted to: Southern California Theory Day USC, November 14, 2015
where \( s_2(N) \) is the sum of digits\(^1\) of the binary representation of \( N \) and \( e_2(N) \) is the exponent of 2 in the \( N \)’s prime factorization\(^2\), or, for \( N \geq 5 \),

\[
2(N - 1) \left( \frac{\lg N - 1}{2} + \varepsilon \right) - 2s_2(N) - e_2(N) + 8 + c,
\]

where \( \varepsilon \), given by:

\[
\varepsilon = 1 + \theta - 2^\theta \text{ and } \theta = \left\lceil \lg(N - 1) \right\rceil - \lg(N - 1),
\]

is a continuous function of \( N \) (briefly analyzed in [1]) that oscillates between 0 and and \( 1 - \lg e + \lg \lg e \approx 0.0860713320559342 \).

The above results allow for deciding if any given \( N \)-element heap is a worst-case heap and if any given \( N \)-element array is a worst-case array for Heapsort, both in \( O(N \log N) \) time.

Keywords: Heap, heapsort, sorting, worst case.

2010 MSC: 68W40 Analysis of algorithms

ACM Computing Classification

Theory of computation: Design and analysis of algorithms: Data structures

design and analysis: Sorting and searching

Mathematics of computing: Discrete mathematics: Graph theory: Trees

Mathematics of computing: Continuous mathematics: Calculus

References


Last modified on November 17, 2015

\(^1\)Equal to the number of 1’s in the binary representation of \( N \).

\(^2\)Equal to the number of trailing 0’s in the binary representation of \( N \).