IMPROVING WORST-CASE BEHAVIOR OF HEAPS

SVANTE CARLSSON

Department of Computer Science, Lund University, Box 725, S-22007 Lund, Sweden

Abstract.

It is possible to improve the worst-case behavior of operations on heaps, when counting number of comparisons between elements in the heap, by modifying the traditional binary heap. The improvement suggested is the use of a heap with "scattered" leaves. This method can be combined with the use of a ternary heap for even greater improvement.

Some other improvements of worst-case behavior compared to standard algorithms are discussed at the end of the paper, where the gain of at least \((N-2)\) comparisons when sorting \(N\) elements with HEAPSORT is of great interest in implementations. An algorithm for inserting a new element in a heap in \(O(\log \log n)\) comparisons between elements is mentioned.

1. Introduction.

The binary heap is a data structure which is used in certain situations, e.g. in priority queues and for the well-known sorting method HEAPSORT. The main reason for this is the good expected and worst-case behavior that comes from it. Therefore it is interesting to compare the worst-case behavior of various types of heaps. We will use the following definitions:

**Definition 1:** A heap is a tree where the element associated with each vertex is greater than or equal to the elements associated with its sons.

From the definition above it follows that the greatest element is associated with the root of the tree, and the sons -- if they exist -- are the roots of other heaps.

**Definition 2:** By adjustment of a heap we mean making a heap of a tree where the sons of the root are roots of heaps, but the root of the tree is not associated with the greatest element. The way this is done is to let the root change place with the son associated with the greatest element, and then adjust the subtree of which the son was a root.

**Definition 3:** The worst-case behavior of a heap means the maximum number of comparisons between elements, needed to adjust a heap.
**Definition 4:** The height of a tree is defined as the number of levels in the tree, where the root will be at level zero.

**2. Improvement by scattering the leaves.**

A complete binary heap stores its elements in the following order, where vertex \( m \) has the sons \( 2m \) and \( 2m + 1 \). The height of a heap with \( n \) elements is \([\log_2 n]\).

Let \( h \) be equal to the height of the heap, then the number of comparisons necessary in the worst case \( (C_n) \) is \( 2h \), except when there is only one single leaf at the lowest level. Then \( C_n = 2h - 1 \).

Hence,

\[
C_n = \begin{cases} 
2[\log_2 n] - 1 & \text{if } n \text{ is a power of } 2 \\
2[\log_2 n] & \text{otherwise.}
\end{cases}
\]

Another possible way to implement a heap is to use a binary heap with scattered leaves, which stores its elements in the following order:

where vertex \( m \) has the sons \( m + a \) and \( m + 2a \), with \( a = 2[\log_2 n] \). The height of a heap with \( n \) elements is \([\log_2 n]\).