heap removal algorithm

Canonical name	HeapRemovalAlgorithm
Date of creation	2013-03-22 12:29:30
Last modified on	2013-03-22 12:29:30
Owner	mps (409)
Last modified by	mps (409)
Numerical id	11
Author	mps (409)
Entry type	Algorithm
Classification	msc 68P20
Classification	msc 68P10
Classification	msc 68P05
Related topic	Heap
Related topic	HeapInsertionAlgorithm
Related topic	Heapsort

Definition

Let H be a heap storing n elements, and let $\preceq$ be an operator that defines a total order over all of the values in H. If H is non-empty, then there is a value x in H such that $x \preceq y$ holds for all values in H. The *heap removal algorithm* removes x from H and returns it, while maintaining the heap property of H.

The process of this algorithm is similar in nature to that of the heap insertion algorithm. First, if H only holds one value, then that value is x, which can simply be removed and returned. Otherwise, let z be the value stored in the right-most leaf of H. Since x is defined by the heap property to be the root of the tree, the value of x is saved, z is removed, and the value at the root of the tree is set to z. Then z is *sifted* downwards into the tree until the heap property is regained.

The sifting process is similar to that of the heap insertion algorithm, only in reverse. First, if z is a leaf, the process ends. Otherwise, let a, b be the two children of z, chosen such that $a \preceq b$ holds. If $z \preceq a$ holds, the process ends. Otherwise, a and z are swapped and the process repeats for z.

Analysis

Since H is a balanced binary tree, it has a maximum depth of $\lceil \log_2 n \rceil + 1$. Since the maximum number of times the sift operation can occur is constrained by the depth of the tree, the worst-case time complexity for heap insertion is $\mathcal{O}(\log n)$.

Pseudocode

What follows is the pseudocode for implementing heap removal. For the given pseudocode, we presume that the heap is actually represented implicitly in an array (see the binary tree entry for details), and that the heap contains at least one value.

HeapRemove($H, n, \preceq$)
Input. A heap $(H, \preceq)$ (represented as an array) containing $n > 0$ values.
Output. Removes and returns a value x from H such that $x \preceq y$ holds for all y in H.
Procedure:

```
$top \leftarrow H[0]$
$n \leftarrow n - 1$
$H[0] \leftarrow H[n]$
$parent \leftarrow 0$
$child \leftarrow 1$
while $child < n$ do
        if $child + 1 < n$
           then
                  if $H[child + 1] \preceq H[child]$
                     then
                            $child \leftarrow child + 1$
                  fi
        fi
        if $H[parent] \not\preceq H[child]$
           then swap($H[parent], H[child]$)
                  $parent \leftarrow child$
                  $child \leftarrow 2 \cdot child + 1$
            else $child \leftarrow n$
        fi
od
```