heapsort

Canonical name	Heapsort
Date of creation	2013-03-22 12:31:02
Last modified on	2013-03-22 12:31:02
Owner	mathcam (2727)
Last modified by	mathcam (2727)
Numerical id	9
Author	mathcam (2727)
Entry type	Algorithm
Classification	msc 68P10
Related topic	HeapInsertionAlgorithm
Related topic	HeapRemovalAlgorithm
Related topic	Heap
Related topic	SortingProblem

The *heapsort* algorithm is an elegant application of the heap data structure to the sorting problem. It consists of building a heap out of some list of n elements, and the removing a maximal value one at a time.

The Algorithm

The following pseudocode illustrates the heapsort algorithm. It builds upon the heap insertion and heap removal algorithms.

Algorithm HEAPSORT$((A, \preceq, n))$

Input: List A of n elements

Output: A sorted, such that $\preceq$ is a total order over A

```
begin
  for $i \leftarrow 2$ton do
        HeapInsert$(A, \preceq, i-1, A[i])$
  for $i \leftarrow n$downto2 do
        $A[i-1] \leftarrow \mathbf{HeapRemove(H, i, \preceq)}$
end
```

Analysis

Note that the algorithm given is based on a top-down heap insertion algorithm. It is possible to get better results through bottom-up heap construction.

Each step of each of the two **for** loops in this algorithm has a runtime complexity of $\mathcal{O}(\log i)$. Thus overall the heapsort algorithm is $\mathcal{O}(n \log n)$.

Heapsort is not quite as fast as quicksort in general, but it is not much slower, either. Also, like quicksort, heapsort is an in-place sorting algorithm, but not a stable sorting algorithm. Unlike quicksort, its performance is guaranteed, so despite the ordering of its input its worst-case complexity is $\mathcal{O}(n \log n)$. Given its simple implementation and reasonable performance, heapsort is ideal for quickly implementing a decent sorting algorithm.