A Brief $\LaTeX$ Example*

I. M. Kool
Denison University, Granville, Ohio 43023
`kool_im@denison.edu`

MAX-HEAPIFY is an important subroutine for manipulating max-heaps. Its inputs are an array A and an index i into the array. When MAX-HEAPIFY is called, it is assumed that the binary trees rooted at LEFT(i) and RIGHT(i) are max-heaps, but that $A[i]$ may be smaller than its children, thus violating the max-heap property. The function of MAX-HEAPIFY is to let the value at $A[i]$ "float down" in the max-heap so that the subtree rooted at index i becomes a max-heap.

MAX-HEAPIFY(A, i)
  $l \leftarrow$ LEFT(i)
  $r \leftarrow$ RIGHT(i)
  if $l \leq$ *heap-size*$[A]$ and $A[l] > A[i]$
    then *largest* $\leftarrow l$
    else *largest* $\leftarrow i$
  if $r \leq$ *heap-size*$[A]$ and $A[r] > A[$*largest*$]$
    then *largest* $\leftarrow r$
  if *largest* $\neq i$
    then exchange $A[i] \leftrightarrow A[$*largest*$]$
      MAX-HEAPIFY$(A,$ *largest*$)$

Figure 1 illustrates the action of MAX-HEAPIFY. At each step, the largest of the elements $A[i]$, $A[$LEFT$(i)]$, and $A[$RIGHT$(i)]$ is determined, and its index is stored in *largest*. ...

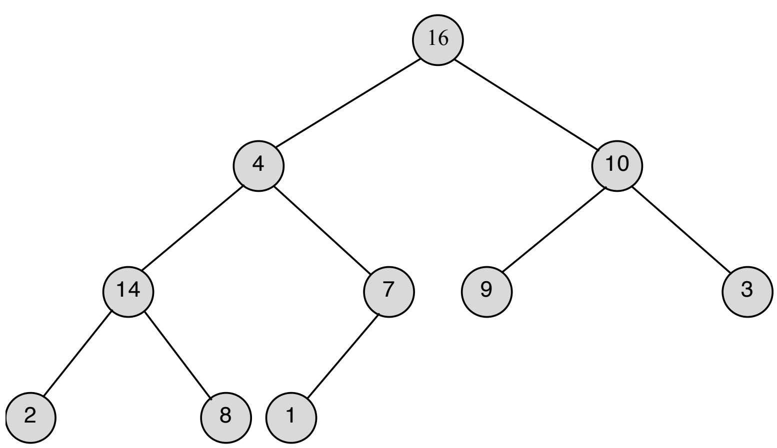

Figure 1: The action of MAX-HEAPIFY$(A, 2)$, where *heap-size*$[A] = 10$. ...

*Taken from pp. 130–131 of *Introduction to Algorithms* by Cormen, et al.