QuickSort

Analysis of HeapSorter()'s running time

Examples of siftdown() and siftup()
Example of stack
\[ (I) \quad \left( \sum_{u \leq \log n} \frac{1}{u} \right)^{0=\eta} = (\eta)O \left( \sum_{u \leq \log n} \frac{1+\eta}{u} \right)^{0=\eta} \]

- The time required by \texttt{steptdown()} when called on a node of height \( \eta \) is \( O(\eta) \).
- The time required is \( O(\eta) \).
- There are at most \( \left\lfloor \frac{n-1}{\eta} \right\rfloor \) nodes of height \( \eta \).
- The tighter analysis relies on the property that in an \( n \)-element heap, there are at most \( \left\lfloor \frac{n-1}{\eta} \right\rfloor \) nodes of height \( \eta \).
- In the tree, and the heights of most nodes are small.
- We can derive a tighter bound than \( O(\eta \log \log n) \).

Analysis of \texttt{HeapSorter()}’s running time.
\[ z = \frac{\varepsilon (2/1 - 1)}{2/1} = \sqrt{\frac{y}{\gamma}} \int_{0}^{\infty} \]

in which \( x_1/2 \) is substituted to yield

\[ \frac{\varepsilon (x - 1)}{x} = \sqrt{\frac{y}{\gamma}} \int_{0}^{\infty} \]

to obtain

\[ \frac{x - 1}{1} = \sqrt{\frac{y}{\gamma}} \int_{0}^{\infty} \]

\( (x > |x| \text{ for } x < 1) \)

by both sides of the infinite geometric series (for \( x < 1 \))

The last summation can be evaluated by differentiating and multiplying

Analysis of HeapSorter()’s Running Time (cont.)
\begin{equation}
(\alpha) O = \left(\frac{y}{\eta} \sum_{n=0}^{\infty} u \right) O = \left(\frac{y}{\eta} \sum_{[u \in \mathcal{L}]_{\eta}} u \right) O
\end{equation}

Thus, the running time of Heapsorter() can be bounded as

\textbf{Analysis of Heapsorter()’s running time (cont.)}
The following diagram illustrate one step.

Quick Sort is a hard-split, easy-join method.
Thus, in the expected case, Quick Sort takes \( O(n \log n) \) steps.

![Diagram of Quick Sort partitioning](image)

- Each element is partitioned into two (almost) equal-sized parts, each element is split \( \log n \) times.

**Quick Sort**

March 20, 2000