Overview

• Quicksort
• Analysis of Heapsorter()’s running time
• Examples of stripper() and stripper()
\[
\left(\frac{y^\frac{1}{2}}{y} \sum_{u \in \log i} u\right) O = \left(\frac{1+y^\frac{1}{2}}{u} \right) \sum_{u \in \log i}
\]

The time required by \texttt{stddympdown()} when called on a node of height \(y\) is \(O(y)\), so we can express the total cost of \texttt{Heapsorter()} as

- The time analysis relies on the property that in an \(n\)-element heap, there are at most \(\left\lfloor \frac{\log n + 1}{\log i} \right\rfloor \) nodes of height \(y\).
- The tighter analysis relies on the property that in the tree, and the heights of most nodes are small.
- In the tree, and the heights of most nodes are small.
- We can derive a tighter bound than \(O(n \log n)\) by observing that the

Analysis of \texttt{Heapsorter()}'s running time.
(4) \[ z = \frac{z(z/2 - 1)}{z/2} = \int_0^\infty y^z e^{-y} dy \]

in which \( z = 1/2 \) is substituted to yield

(3) \[ \frac{z(x - 1)}{x} = \int_0^\infty y^x e^{-y} dy \]

to obtain

(2) \[ \frac{x - 1}{1} = \int_0^\infty y^x e^{-y} dy \]

(by both sides of the infinite geometric series (for \( 1 > |x| \)) by differentiating and multiplying

The last summation can be evaluated by differentiating and multiplying

Analysis of Heapsorter's running time (cont.)
\[(u) \mathcal{O} = \left( \frac{y^2}{y} \sum_{0=y}^{\infty} u \right) \mathcal{O} = \left( \frac{y^2}{y} \sum_{[u \leq 0]}^{\infty} u \right) \mathcal{O}\]

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

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.

• If the pivot chosen by split() divides the array into two (almost) equal-sized parts, each element is split $\log n$ times.