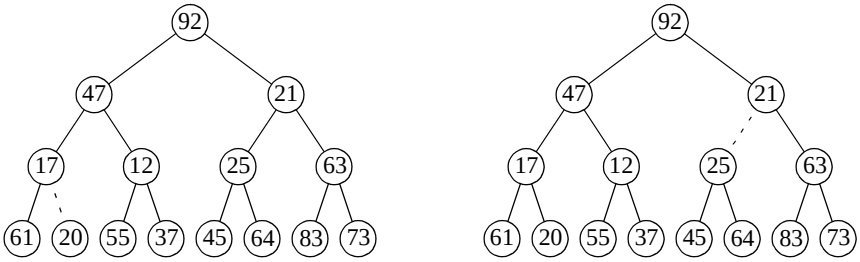

Figure 20.22 After `percolateDown(4)` (left); after `percolateDown(3)` (right)

To bound the running time of `fixHeap`, we must bound the number of dashed lines. This can be done by computing the sum of the heights of all the nodes in the heap, which is the maximum number of dashed lines. We expect a small number because half of the nodes are leaves and have height zero and a quarter have height 1. Thus only a quarter of the nodes (those not already counted in the first two cases) can contribute more than 1 unit of height. In particular, there is only one node that contributes the maximum height of $\lfloor \log N \rfloor$.

The linear time bound can be shown by computing the sum of the heights of all the nodes in the heap.

To obtain a linear time bound for `fixHeap`, we need to establish that the sum of the heights of the nodes of a complete binary tree is $O(N)$. This is shown in Theorem 20.1. We prove the bound for perfect trees by using a marking argument.

We prove the bound for perfect trees by using a marking argument.

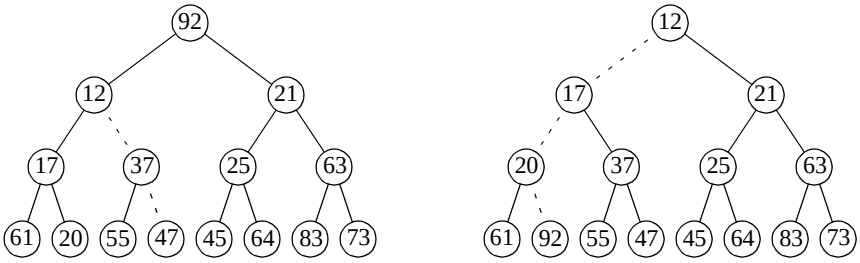

Figure 20.23 After `percolateDown(2)` (left); after `percolateDown(1)` and `fixHeap` terminates (right)