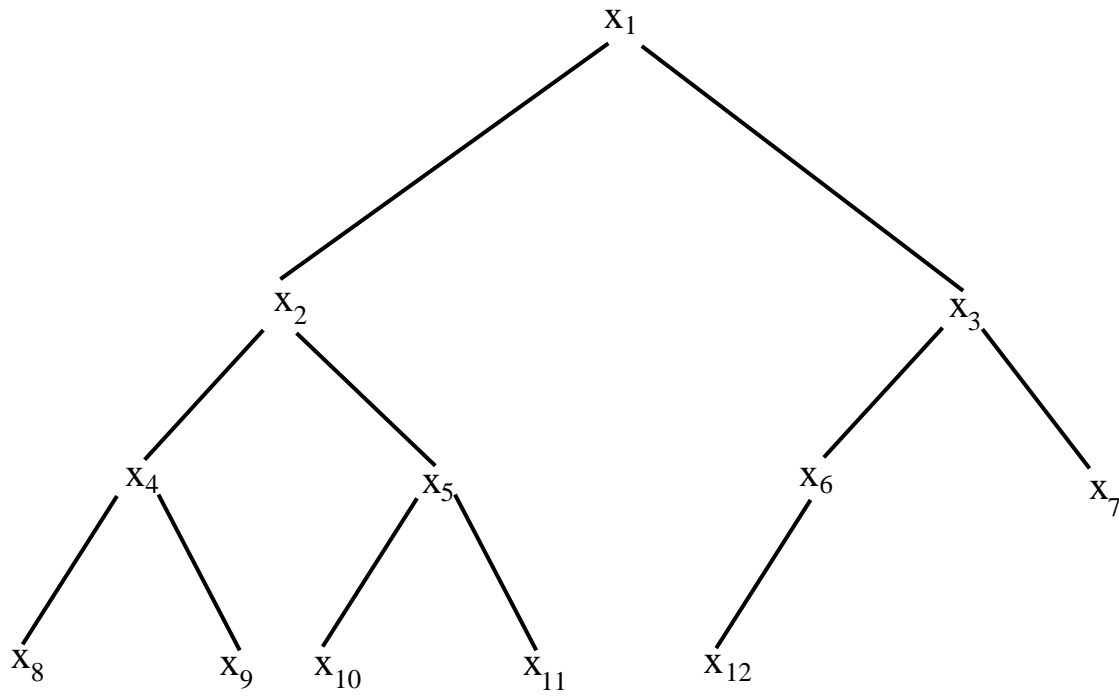


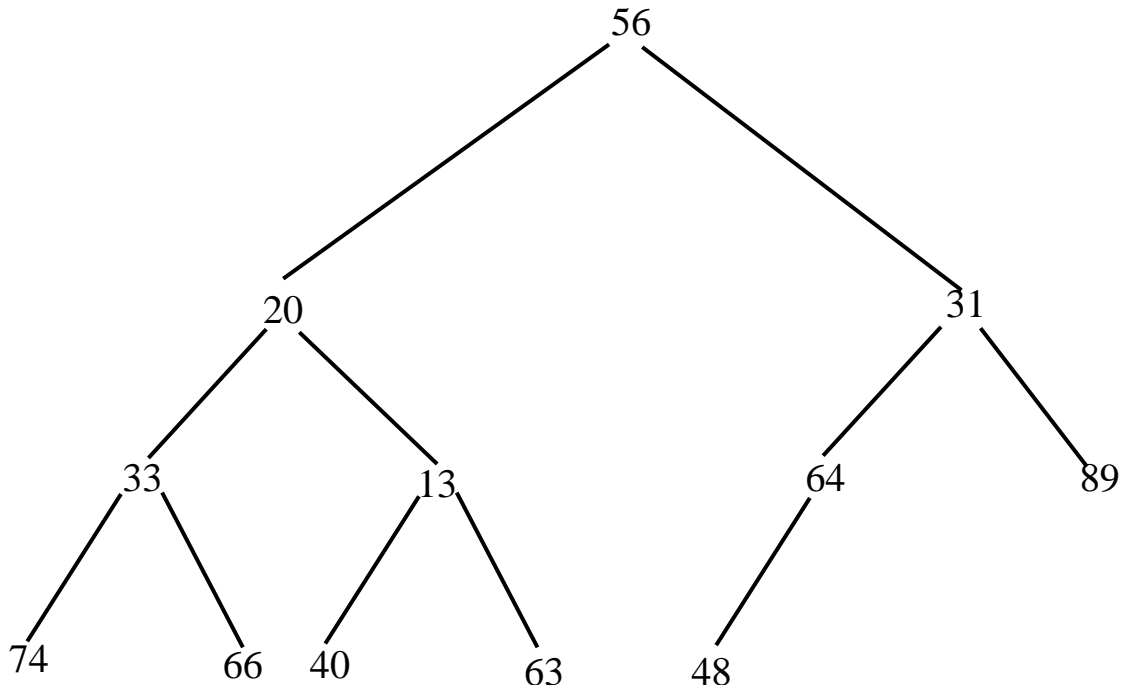
# Heapsort

Peter Young

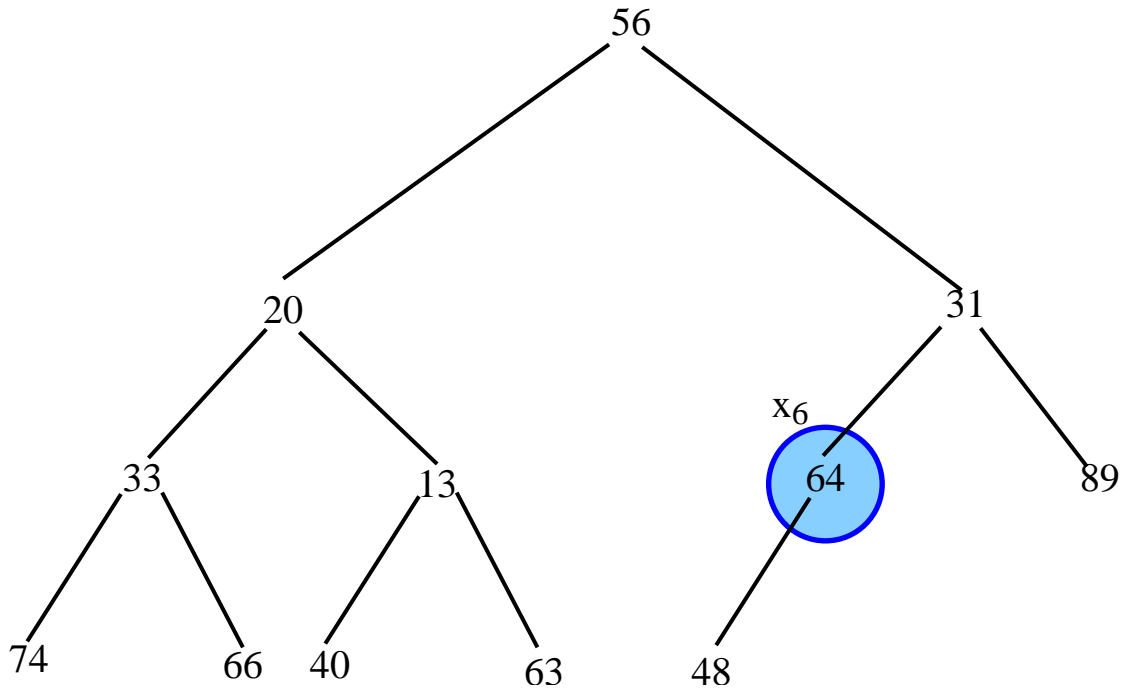




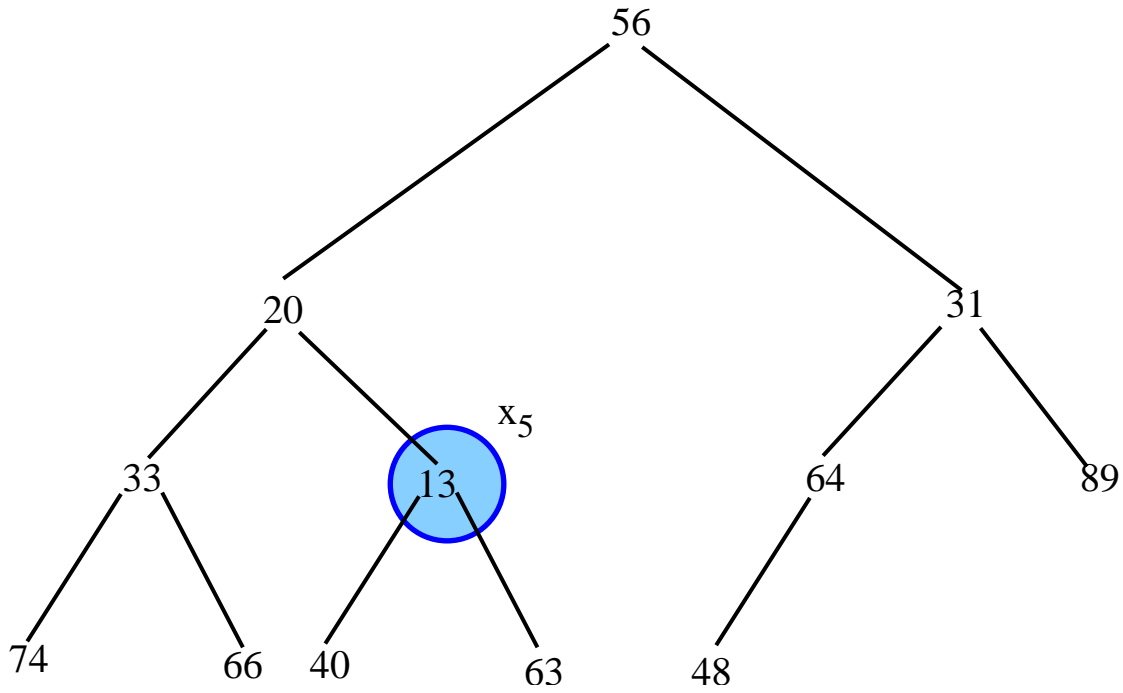
The data is put on binary tree.



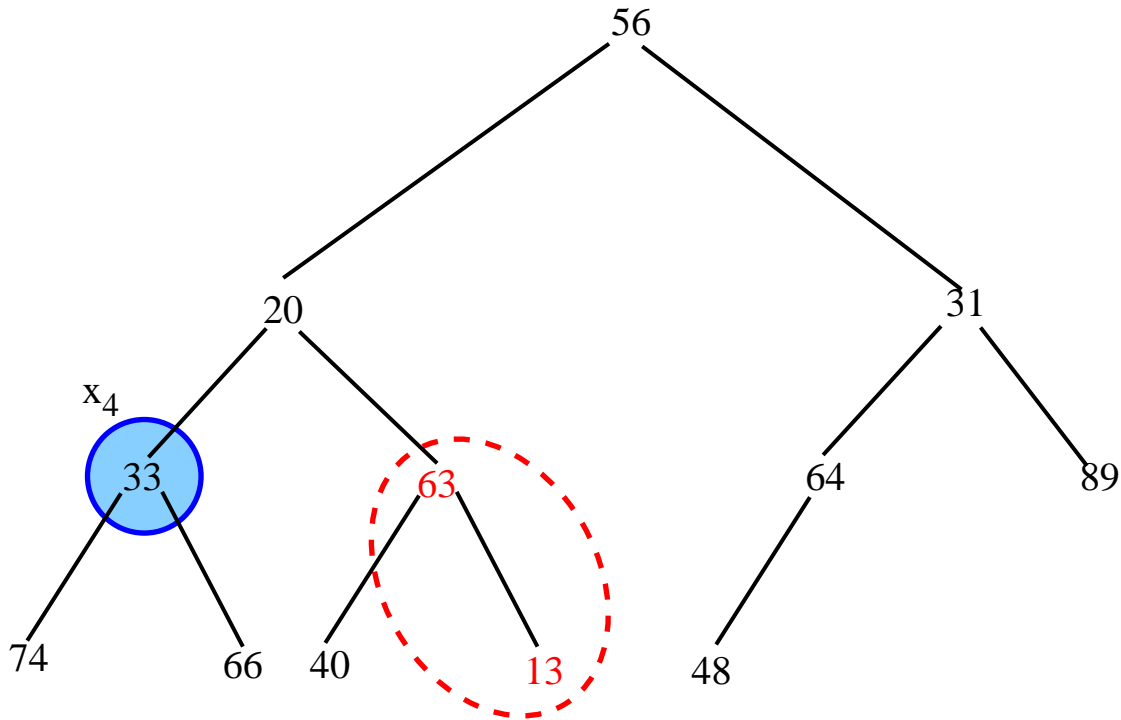
**The original (unsorted) array.**



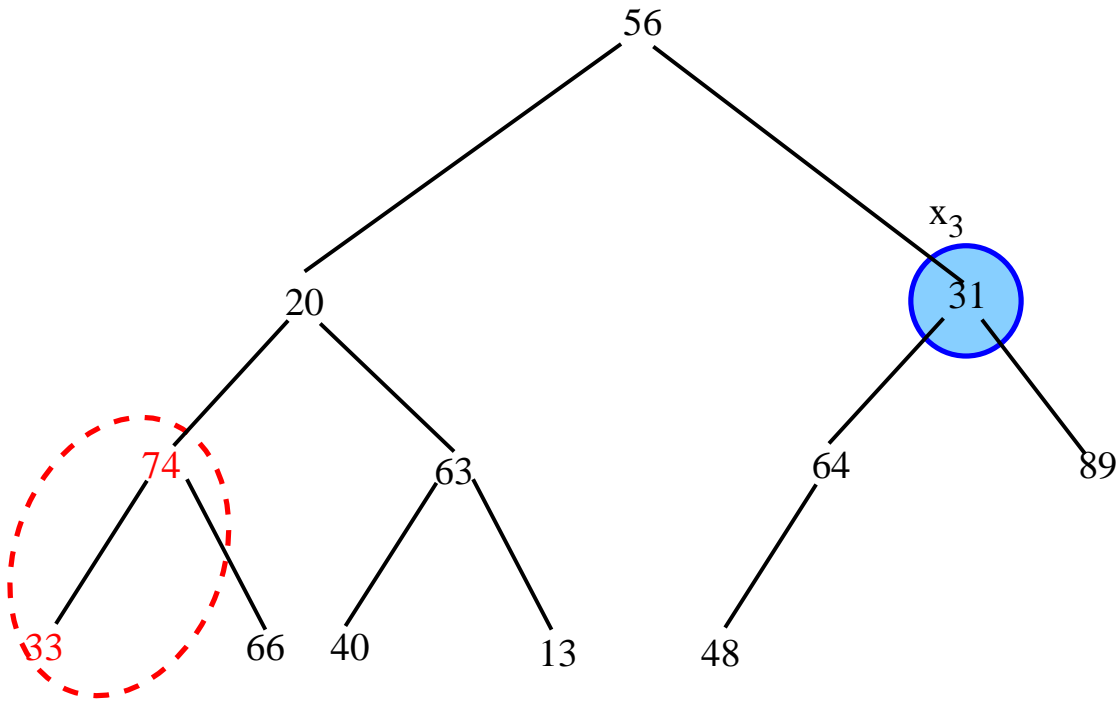
We start to form the heap, and go first to element  $\lfloor n/2 \rfloor$ . Here  $n = 12$  and element 6 has value 64. It is larger than the number below it so it stays.



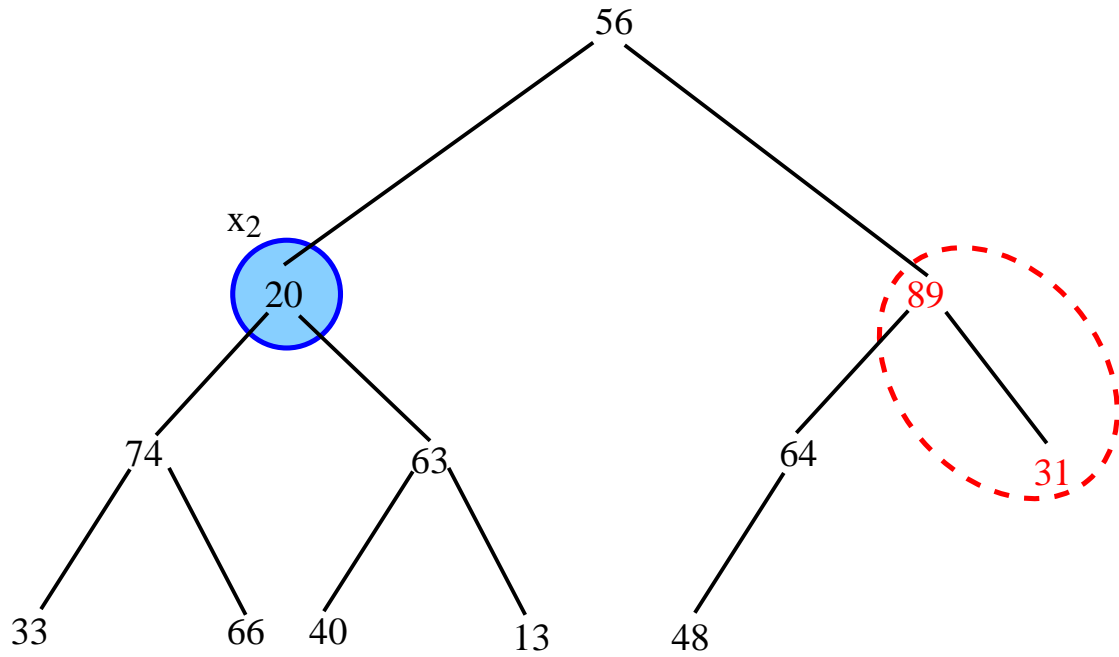
Next go to  $x_5 = 13$ . We will swap this with 63 (the larger of the two below it).



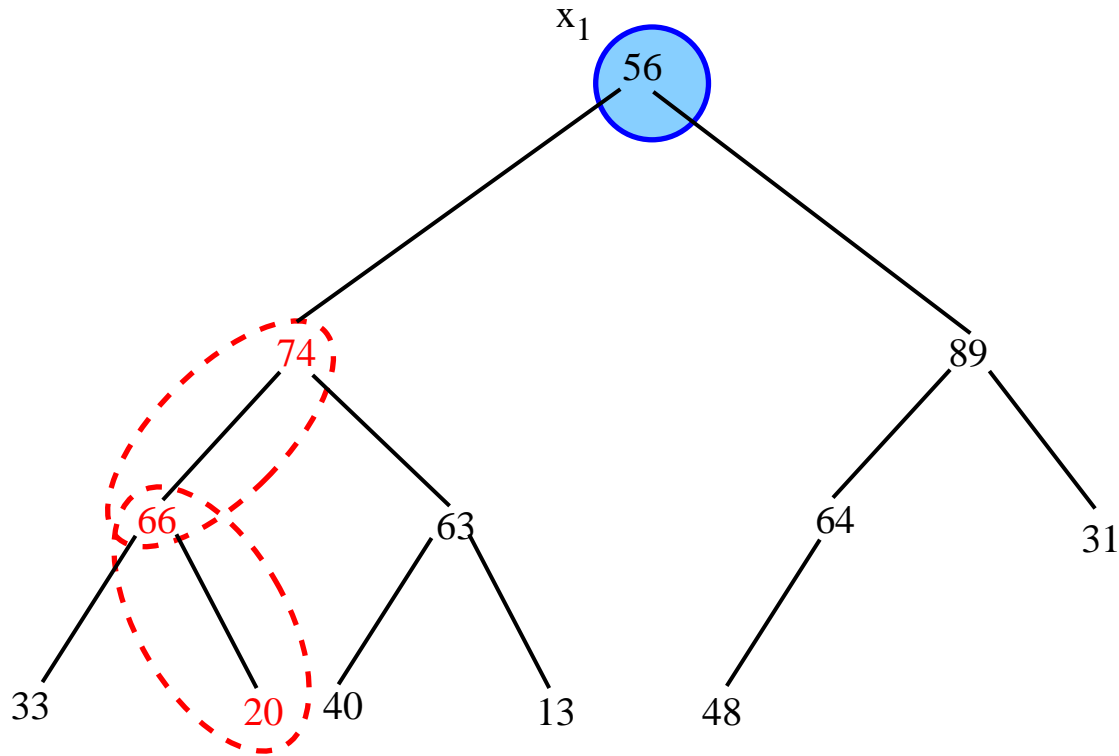
The dashed oval indicates that 13 and 63 have been swapped. The solid circle indicates that we now go to  $x_4 = 33$ . We will swap this with 74 below it.



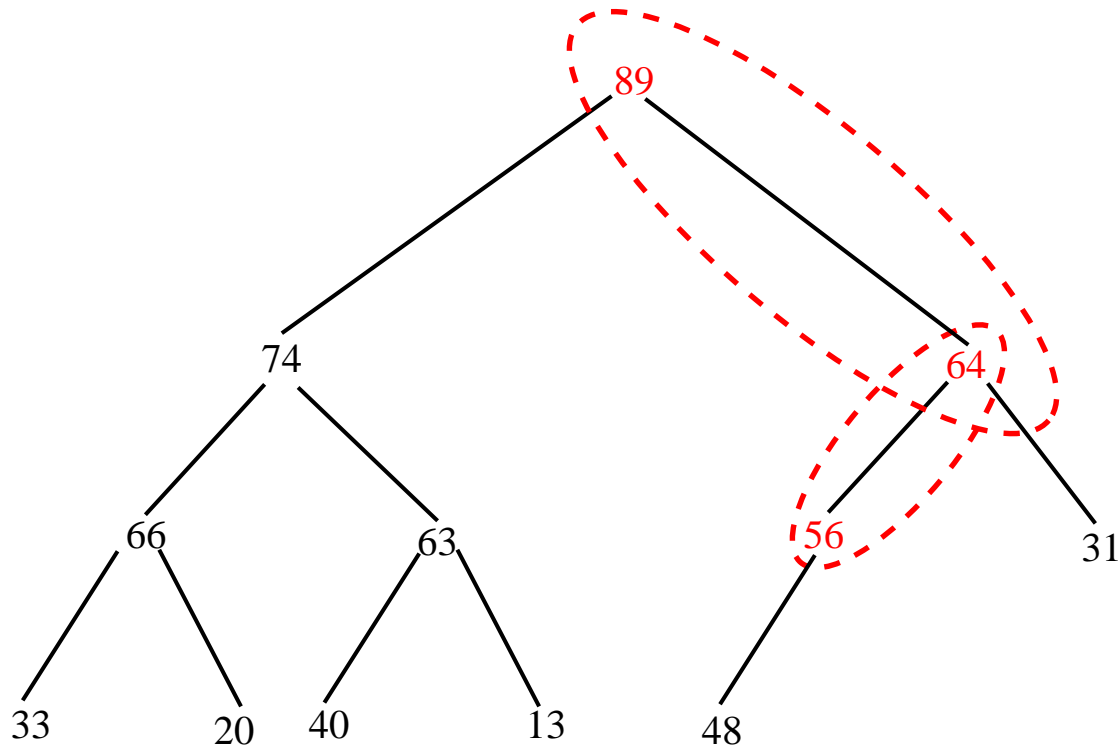
74 and 33 have been swapped. We now go to  $x_3 = 31$ . This will be swapped with 89.



89 and 31 have been swapped. We now go to  $x_2 = 20$ . We will swap this with 74 (the larger of the two below it) and then with 66 (the larger of the two below its new position).

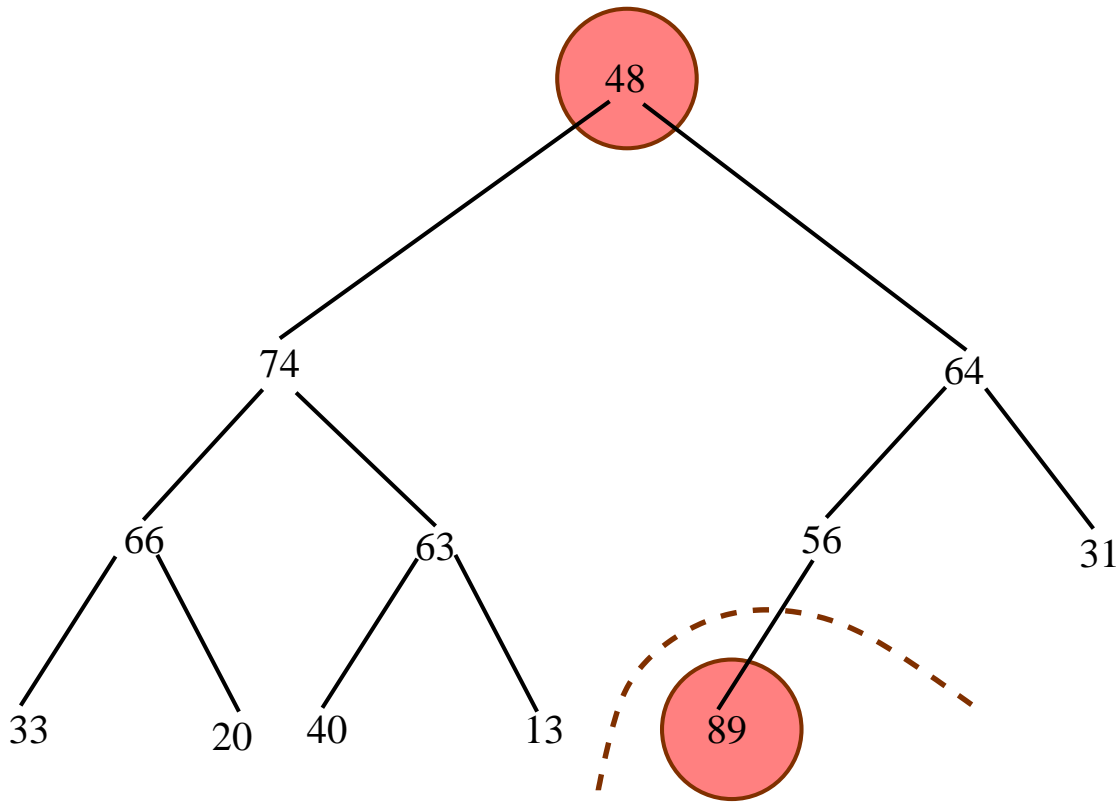


20 has been pulled down to the bottom raising up 66 and 74. We now go to  $x_1 = 56$ . This will be swapped with 89 and then with 64. (It does not continue down to the bottom row because  $56 > 48$ ).

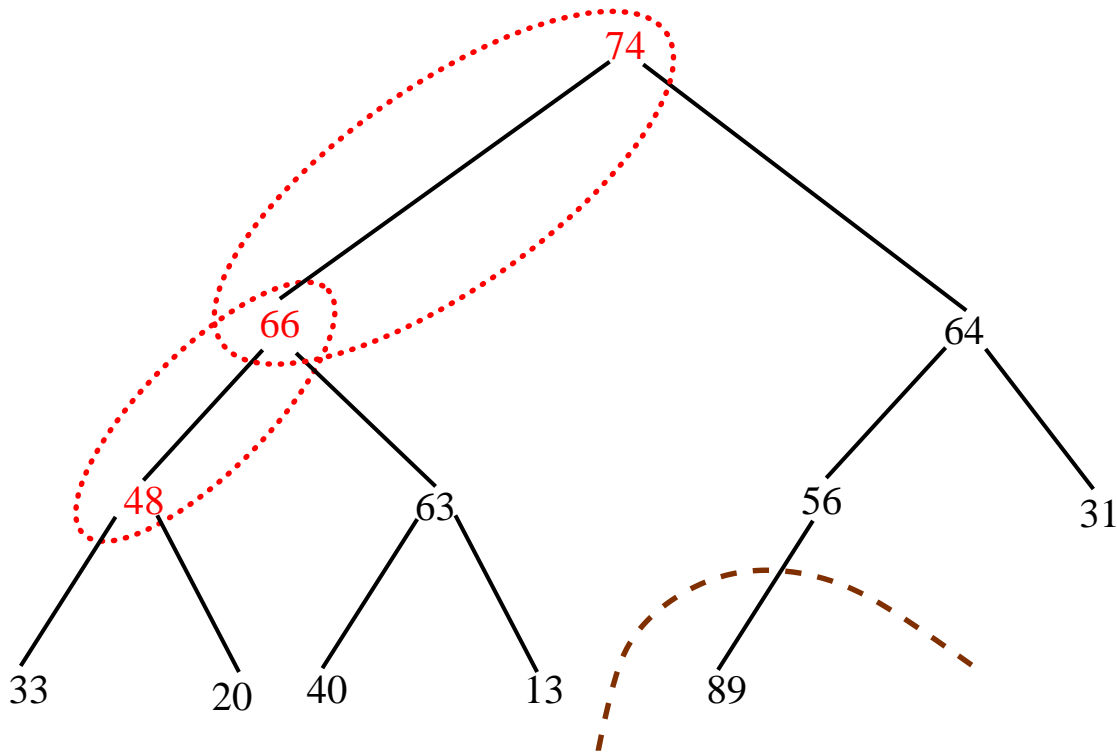


56 moved down two rows, and 64 and 89 moved up.

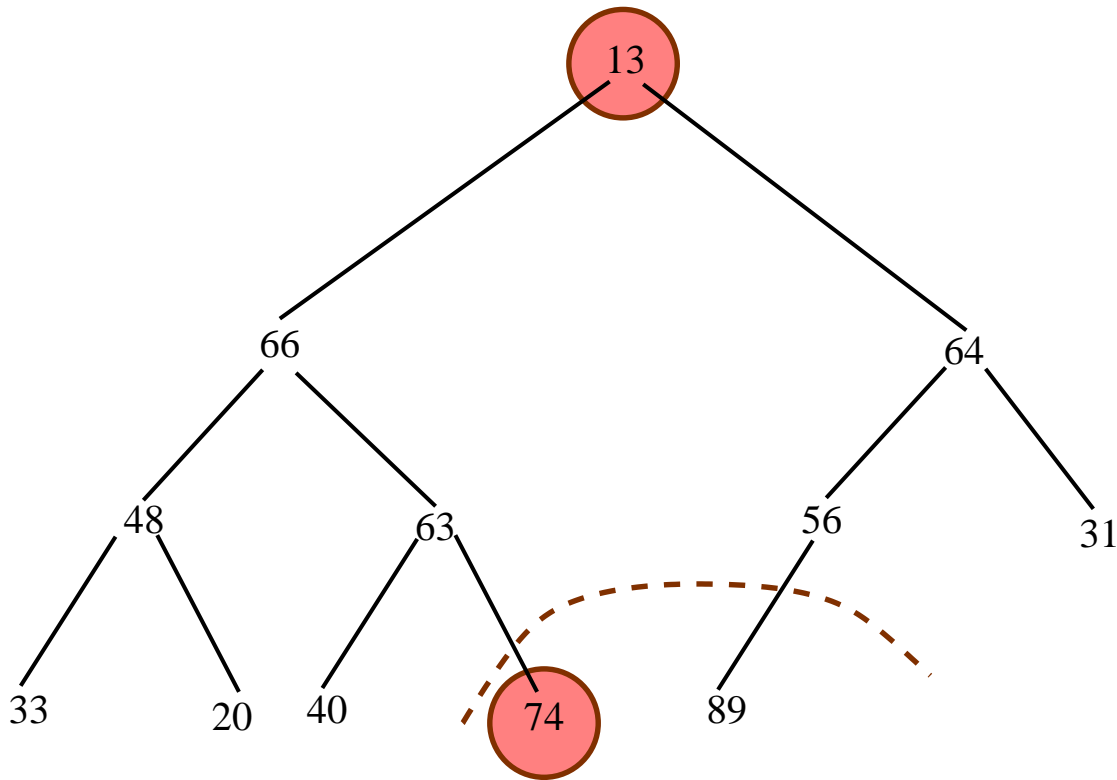
**The heap has now been formed.**



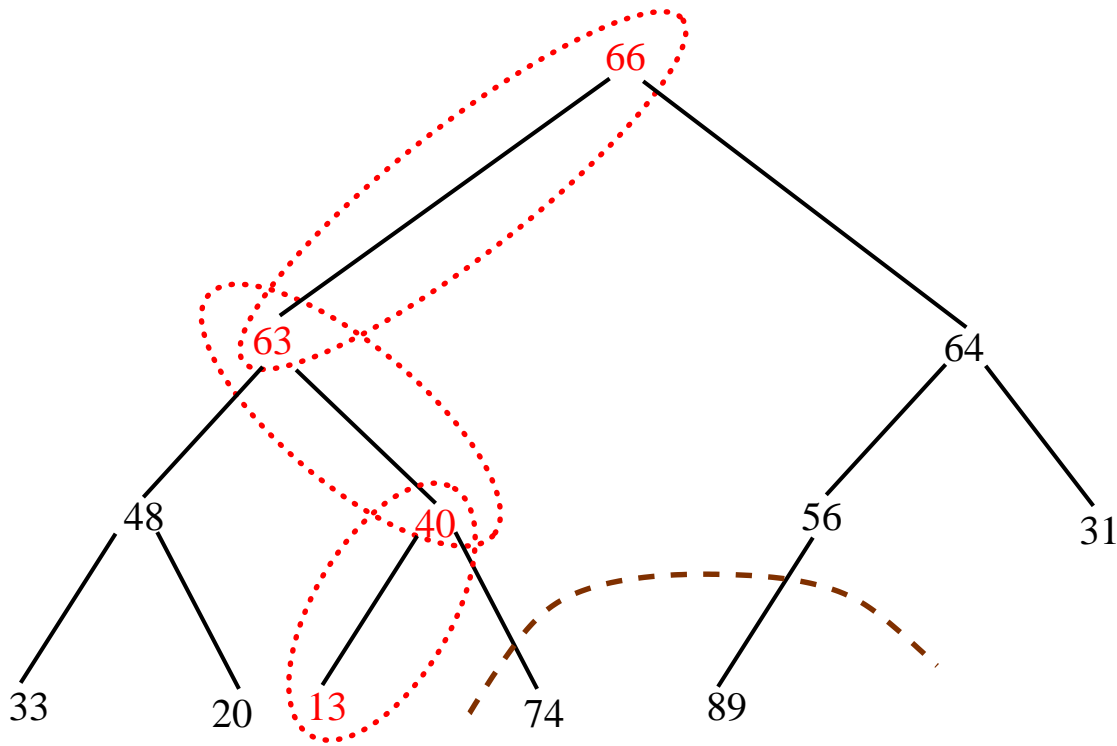
The top element in the previous figure (89) (the largest element) is interchanged with the element in the last node in the previous figure (48). The last node, now containing the largest element, is then made inactive.



The top element in the previous figure (48) has been sifted down to its correct location, thus raising up 66 and 74. The latter is now the largest active element and so is on top.

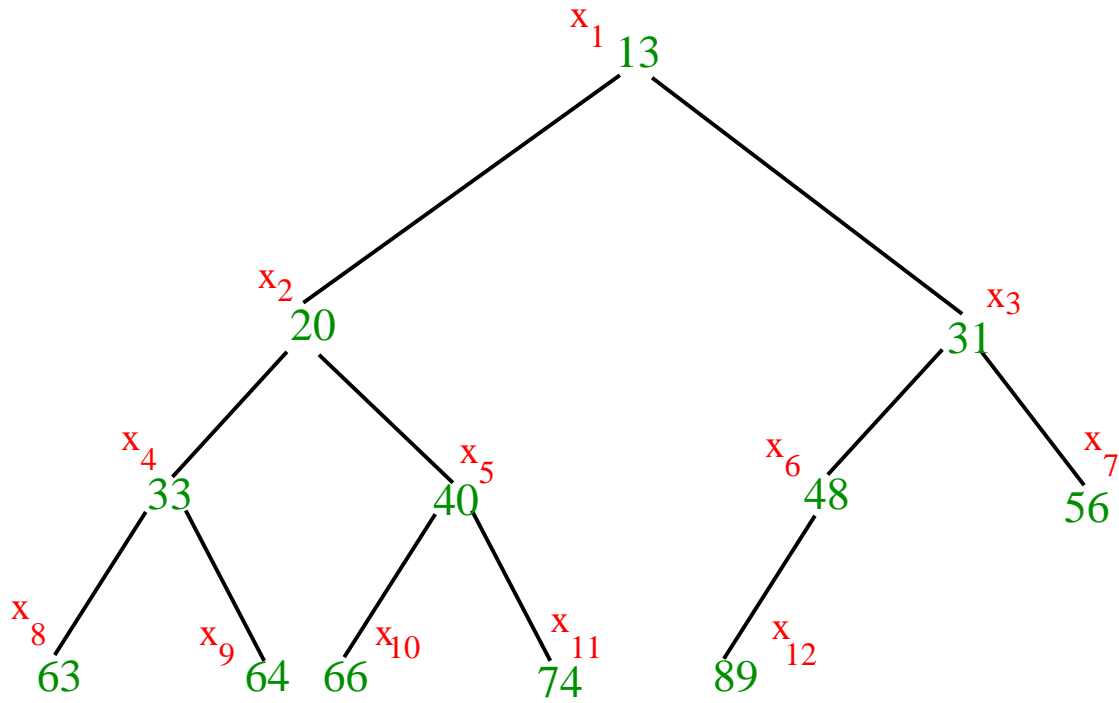


The top element in the previous figure (74) (which is the largest active element) is interchanged with the element in the last active node in the previous figure (13). This last node, now containing the second largest element, is then made inactive.



The top element in the previous figure (13) has been sifted down to its correct location, and the largest active element (66) is thereby brought to the top.

This process repeats 9 more times to give ...



**The sorted array.**