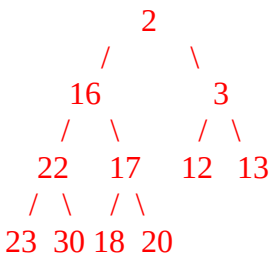
2) (5 pts) ALG (Binary Heaps)

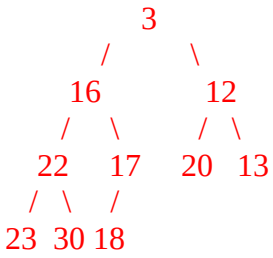
The array below stores a minimum binary heap. Draw the tree version of the corresponding binary heap. Then, remove the minimum value and show the resulting heap, in tree form. (Note: Index 0 isn't shown because index 1 stores the value at the root/top of heap.)

Index	1	2	3	4	5	6	7	8	9	10	11
Value	2	16	3	22	17	12	13	23	30	18	20

Here is the initial heap, in tree form:



When we delete the minimum, 2, stored at the top, 20, the value in the "last" location replaces it (to maintain the structural integrity of the heap.) From there, we percolate 20 down, swapping it with 3, and then 12 to get the resulting tree:



Grading: 2 pts for correct drawing, 1 pt is something minor is off, 0 otherwise.
1 pt if structural location of 20 is removed, 1 pt for incorrect percolateDown,
2 pts for correct percolateDown. (If the drawing is significantly wrong, don't give any credit for the second part. If it's slightly wrong, map points as best as possible.)