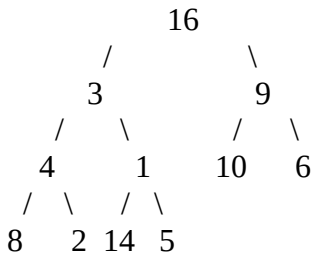
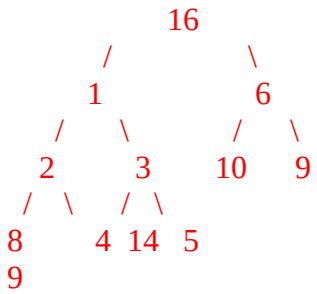
**2)** (10 pts) ALG (Binary Heaps)

Consider running the Make Heap/Heapify algorithm on the following set of random values (shown stored in a heap structure) to convert it to a min heap. In doing so, exactly six swaps will occur between adjacent values in the heap. When the algorithm concludes, the structure will be valid minheap. Draw the structure of the tree after (a) the first three swaps have completed, and (b) when the Make Heap algorithm completes (the final valid minheap). Here is the initial tree drawing of the values:



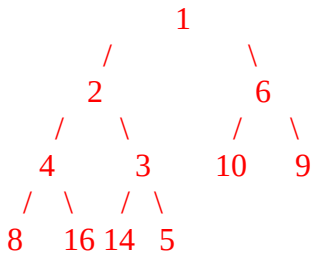

(a) Draw the minheap here after completing the first three swaps.

(a) After we run percolate down on the node initially storing the 1, then the 4, then the 9 and then the 3, we make three swaps: (4, 2), (9, 6) and (3, 1). Here is the picture of the heap after those swaps:




(b) Draw the final minheap.

(c) Then, we run percolate down on the root node and it will successively swap with 1, 2 and 4 producing the following result:




**Grading: 2 pts for swap of 4 and 2 in first pic,**
**2 pts for swap of 9 and 6 in first pic,**
**2 pts for swap of 3 and 1 in first pic,**
**1 pt each for placement of 1, 2, 4 and 16 in second pic**