Suppose we base our max-heaps on ternary instead of binary trees. Each node now has a maximum of three children. We still require the tree to be filled by levels: the root, then the three left-to-right children of the root, then the nine left-to-right children at level 2, and so forth. A tree with \( n \) nodes will always have all levels completely full, except possibly the bottom level. For the questions below, assume the tree has \( n \) nodes and levels 0, 1, 2, \ldots, \( k \). For working out the time complexities required, you can assume that all levels are full. We modify the max-heap property to read “A tree satisfies the max-heap property if the value at each node is no smaller than the values at its children.”

1. Prove that \( k = \Theta(\log_3 n) \).

2. Explain how to map the tree into an array with indices running from 1 to \( n \). What formulas calculate the indices of the left, middle, and right children as a function of the parent index? What formula calculates a parent index from the index of a child?

3. Modify the MaxHeapify algorithm for binary heaps to work for ternary heaps. That is, the new routine MaxHeapify(A, i) rearranges the contents of the nodes in the subtree rooted at index \( i \) to satisfy the max-heap property, assuming that the subtrees rooted at the children of index \( i \) already satisfy the property. Show the the worst-case time complexity of the algorithm is \( \Theta(\log_3 n') \), where \( n' \) is the number of nodes in the subtree rooted at \( i \).

4. Modify the BuildMaxHeap algorithm to work for the ternary case. The new algorithm accepts an array with no constraints on the size relationships among the entries. It considers the array to be a ternary tree and by judiciously applying the new MaxHeapify algorithm, it converts the array into a heap. That is, the final array, interpreted as a ternary tree, satisfies the max-heap property at each node. Show the the new algorithm is \( \Theta(n) \), worst case.

5. Finally, modify the Heapsort algorithm to work in the ternary case. Show that the worst-case time complexity is \( \Theta(n \log_3 n) \).