

2. Heaps (13 pts.)

Given the following array numbers: [4, 2, 7, 6, 12, 9, 1, 10, 8] (the number 4 is in position 0 and 7 in position 2), find the **binary min-heap** that results from calling the make-heap (build-heap) algorithm that uses *heapify* on the array and that runs in linear time.

In the area below show the final array contents (similar to the format we showed above). For full credit, you must also show your work at each step in the algorithm **by drawing the binary tree representation** (not array form, but tree form) after each heapify step until the algorithm is complete. *For each call, you need only show the relevant part of the tree that heapify operates on. Clearly show what argument/index you are calling heapify() on for each step.*

Hint: You don't need to call heapify on EVERY node but can start at a certain location.

Final array contents after the algorithm completes:

[__, __, __, __, __, __, __, __, __]

Diagrams of intermediate trees for each heapify call (**indicate what location you start your heapify call on (e.g. heapify(0)) and then show your diagram of the relevant part of the tree for that operation.**