1. Recall that $f(n)$ is $\Theta(g(n))$ if $f(n)\epsilon O(g(n))$ and $g(n)\epsilon O(f(n))$. Give a closed-form solution in terms of Θ for the following recurrences. Also, state whether the recurrence is dominated at the root, the leaves, or equally at all levels of the recurrence tree. You do not have to show your work, but it might help you get partial credit.

   (a) (5%) $f(n) = 5f(n/5) + \Theta(n)$

   (b) (5%) $f(n) = 3f(n/2) + \Theta(n^2)$

   (c) (5%) $f(n) = f(n/2) + \Theta(\log n)$

   (d) (5%) $f(n) = 5f(n/8) + \Theta(n^{2/3})$

2. (6%) Draw a binary search tree that has the following post-order traversal:

   $$5 \quad 21 \quad 26 \quad 38 \quad 20$$

3. Following are three traversals produced by the exact same binary search tree. Using your powers of inference, determine which one is which. (Fill in each of the blanks with "in-order," "pre-order," or "post-order.")

   (a) (3%) _____ traversal:　　8　5　4　-5　6　12　27　25　12　20

   (b) (3%) _____ traversal:　　-5　4　6　5　20　13　25　27　12　8

   (c) (3%) _____ traversal:　　-5　4　5　6　8　12　13　20　25　27

   The sorted traversal must be the in-order traversal. In pre-order, the root is always printed first. In post-order, the root is always printed last. Using that knowledge, we can infer that 8 must be the root; it's the only number that appears in both the first and last position among those traversals.

4. (15%) The given heap is stored in the array:

94	80	50	35	66	21	13	25

   illustrate all procedures for the ascending heap sort.

5. Transform the prefix expression ++A−∗∗GCT/+AG∗CTA to the following:

   (a) (5%) infix expression

   (b) (5%) postfix expression

   and (c) (5%) draw its expression tree.

6. Draw an AVL tree that satisfies the following three conditions:

   (a) (5%) The tree has exactly 11 nodes.

   (b) (5%) There are no pair of nodes that, if extracted one after the other, will cause the height decrease by 1.

   (c) (5%) There is no key whose insertion will increase the height by 1.

7. (20%) Consider the following undirected, weighted graph:
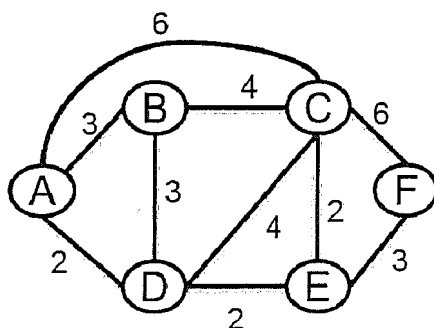
   Step all procedures through Dijkstra's algorithm to calculate the single-source shortest paths from node A to every other nodes.