ch5

【1】

A: Level 1, Non-Leaf Node (root of tree)

B: Level 2, Non-Leaf Node

C: Level 3, Non-Leaf Node

D: Level 3, Leaf Node

E: Level 4, Leaf Node

【2】

- Preorder: A, B, C, D, E
- Inorder: E, D, C, B, A
- PostOrder: E,D, C, B,A
- LevelOrder: A, B, C, D, E

- Preorder: A, B, D, H, I, E, C, F, G
- Inorder: H, D, I, B, E, A, F, C, G
- PostOrder: H, I, D, E, B, F, G, C, A
- LevelOrder: A, B, C, D,E,F,G,H, I
-

【3】

- Preorder: A, B, C, E, D
- Inorder: E, C, B, D, A
- PostOrder: E, C, D, B, A
- LevelOrder: A, B, C, D, E

【4】

The Max Priority Queue is listed in array order:

- Insert 7 : 7
- Insert 16: 16, 7
- Insert 49: 49, 7, 16
- Insert 82: 82, 49, 16, 7
- Insert 5: 82, 49, 16, 7, 5
- Insert 31: 82, 49, 31, 7, 5, 16
- Insert 6: 82, 49, 31, 7, 5, 16, 6
- Insert 2: 82, 49, 31, 7, 5, 16, 6, 2
- Insert 44: 82, 49, 31, 44, 5, 16, 6, 2, 7

(b) The Min Priority Queue is listed in array order:

- Insert 7 : 7
- Insert 16: 7, 16
- Insert 49: 7, 16, 49
- Insert 82: 49, 16, 7, 82
- Insert 5: 5, 7, 49, 82, 16
- Insert 31: 5, 7, 32, 82, 16, 49
- Insert 6: 5, 7, 6, 82, 16, 49, 32
- Insert 2: 2, 5, 6, 7, 16, 49, 32, 82
- Insert 44: 2, 5, 6, 7, 16, 49, 32, 83, 44

【5】

```
int searchHeap(int x)

{/* search for x in the heap
   Time if O(n) since heaps aren't
  organized for searching */
  int i;
  for (i=1; i<=n; i++)
    if (heap[i].key == x) return i;
  return 0;
}
```

【6】

The transformation of a tree into a forest can be defined as follows: If T is a Tree with Root R, then the Forest F corresponding to this tree is denoted by T1 .. Tn+1: (1) is empty, if R is null. (2) T1 has root equal to R, and children equal to C1 .. Cn, where C1 .. Cn are the children of R in T. (3) T2 .. Tn+1 are trees consisting of the descendants of C1 .. Cn in T. Tk is empty if Ck has no descendants in T. The transformation is unique only in that T2 must match the descendants of C1, and so on. The initial ordering of the children in T is irrelevant.

【7】 The correspondence between the PreOrder traversal of a binary tree and its equivalent forest representation is fairly obvious if we look at the linked list representation of the tree. For example, assume that we have the following binary tree:

```
              A
            /   \
           B      C
         /  \    /  \
        D     E F     G
```

The linked representation of this tree would be: (A(B(D,E)C(F,G)). This would produce the following forest of trees:

```
              A                  D                  F
             / \                 |                  |
            B   C                E                  G


             T0                  T1                 T2
```

The PreOrder traversal of the binary tree produces the sequence: ABDECFG. Notice that this corresponds to the list representation. For the forest, we follow A (Node), B (left child), Tree T1, C (right child), and Tree T2.

【8】

You will probably want to refer to Figure 5.49 of the text during this discussion. This figure shows the five legal trees obtained by permuting three nodes. Problem 3.1 gave the formula for determining the number of permutations, and, hence, the number of distinct binary trees. Here we'll just elaborate by examining the trees appearing in Figure 5.49.

As the figure shows, the PreOrder traversal is always 1,2,3. However, the InOrder traversals vary widely. These traversals are, from left to right: 123, 132, 213, 231, 321. We can easily reproduce these results by using the railtrack analogy discussed in problem 3.1. Let me indicate how each of the InOrder traversals was obtained using this analogy.

Tree 1: Each number is added to the stack and then removed from the stack. The stack contains no more than one element at any given time.

Tree 2: The first number is added to, and removed from, the stack. We then add the remaining two numbers as a unit, and remove each of them.

Tree 3: The first two numbers are added to the stack as a unit. Both numbers are removed before the last number is added.

Tree 4: The first two numbers are added to the stack. Only the second number is removed. The third number is added to the stack, and the remaining two numbers are removed.

Tree 5: The three numbers are added to the stack as a unit. They are then removed individually.

Because the traversals follow the stack's operations certain combinations are illegal. For example, it is impossible to arrive at the combination 312. Why? Because there is no way we could remove three from the stack followed by a one.