Problem 1

True or false: (You need not explain your answer. However, in using the sample exam to study with, you should make sure that you understand the reason for the answer.)

A. The height of a heap of size $N$ is always approximately $\log_2(N)$.
B. The height of a binary search tree of size $N$ is always approximately $\log_2(N)$.
C. If you have an unsorted array of size 1000, heapsort will definitely run faster than insertion sort.
D. If you have an unsorted array of size 1000, quicksort will definitely run faster than heapsort.
E. Suppose you have a hash function $h$ from Strings to Integers. Let $a$ and $b$ be two strings. Then $h(a) < h(b)$ if and only if $a$ precedes $b$ alphabetically.

Problem 2:

Give the preorder, postorder, and breadth-first order of the following tree.

Problem 3:

Suppose that you have a binary tree of the following structure:
A. Put the numbers [2,5,9,12,16,19,22,25,28] into this tree so that it forms a binary search tree.

B. Show the final result if you execute add(20) and delete(16) in sequence on this tree.

Problem 4:

The fact that prefix and postfix expressions are unambiguous depends on the fact that each operator has a fixed number of arguments. Most programming languages, however, use the minus sign — both as a binary operator, as in $7 - x$, and as a unary operator, as in $-x$. Give an example to illustrate that, if you use the minus sign in both these ways, then prefix notation is no longer unambiguous. That is, you want to give a string that can be read as valid prefix notation in two different ways with two different values. (Hint: There is a solution with only four symbols.)

Problem 5:

Suppose you have a hash table class with methods $\text{H.put}(\text{T key}, \text{U value})$, and $\text{H.get}(\text{T key})$. Here $\text{T}$ and $\text{U}$ are generic classes. Also, suppose you have a generic singly linked list class defined as follows

```java
class LL<T> {
    public T value;
    public LL<T> next;
}
```

Write in Java a static method $\text{boolean subset}(\text{LL<String> x, LL<String> y})$. The input parameters $x$ and $y$ are unordered linked lists with header, which may have repeated element. The method should return $\text{true}$ if every value in $x$ occurs in $y$; how many times it occurs is unimportant.

For instance if $x$ is the list of Strings ["A", "B", "C", "B", "A"] and $y$ is the list ["C", "B", "D", "C", "A"] then $\text{subset}(x,y)$ should return $\text{true}$.

You should write the code so that it takes advantage of the hash table to work efficiently. Specifically, you will get full credit if the code runs in time proportional to the sum of the lengths of $x$ and $y$. If it runs in time proportional to $|x| \cdot |y|$ then you will get 2/3 credit (for that, you don’t need the hash table).