You have 50 minutes. You may use one prepared 8.5 × 11 sheet of paper. All questions concern the C++ programming language. Check your answers.

For the multiple choice questions (marked [MC]) circle the letter of the best answer, even if no answer is ideal. For the other questions, write the answer after the question in a clear and readable way.

1. [MC] If \( x = \log_2(60) \), then
   (a) \( 30 < x < 60 \)
   (b) \( 4 < x < 5 \)
   (c) \( 5 < x < 6 \)
   (d) \( 6 < x < 7 \)
   (e) \( 7 < x < 8 \)

2. [MC] Only one of the following is true. Which one?
   (a) \( n^2 \) is \( O(n) \)
   (b) \( n^2 + 5n \) is \( O(n^2) \)
   (c) \( n^3 \) is \( O(n^2 + 3n + 10) \)
   (d) \( n^3 \) is \( O(n^2 + 10n) \)
   (e) \( n \) is \( O(\log(n)) \).

3. [MC] To within a constant factor, how long does it take, in the worst case, to insert a value into a height-balanced binary search tree that has \( n \) values in it?
   (a) \( \Theta(1) \)
   (b) \( \Theta(\log_2(n)) \)
   (c) \( \Theta(n) \)
   (d) \( \Theta(n \log_2(n)) \)
   (e) \( \Theta(n^2) \)
4. [MC] To within a constant factor, how long does it take, in the worst case, to insert a value into a min-heap that has \( n \) values in it?

(a) \( \Theta(1) \)
(b) \( \Theta(\log_2(n)) \)
(c) \( \Theta(n) \)
(d) \( \Theta(n \log_2(n)) \)
(e) \( \Theta(n^2) \)

5. [MC] To within a constant factor, how long does it take, on the average, to look up a value in a hash table that has \( n \) values in it, assuming a high quality implementation?

(a) \( \Theta(1) \)
(b) \( \Theta(\log_2(n)) \)
(c) \( \Theta(n) \)
(d) \( \Theta(n \log_2(n)) \)
(e) \( \Theta(n^2) \)

6. [MC] Suppose that you start with an empty priority queue that is implemented using a min-heap. You successively insert \( n \) different values, with their priorities, into the priority queue. How much time does it take, in the worst case, to do all of the insertions? (Give the cumulative time for all insertions, not just the time for one of them.)

(a) \( \Theta(1) \)
(b) \( \Theta(\log_2(n)) \)
(c) \( \Theta(n) \)
(d) \( \Theta(n \log_2(n)) \)
(e) \( \Theta(n^2) \)
Min-heap $H_0$ is as follows.

```
        3
       / \   \   \
   12   6    \
  /     /     \
17  14  9  20
```

7. If you insert 2 into min-heap $H_0$ and then restore the heap order, what heap results?

8. If you remove the smallest value from min-heap $H_0$ and then restore the heap order, what heap results? Start with the heap shown above, not the result that you got from problem 7.
Consider the following binary search tree, $T_0$. 

![Binary Search Tree Diagram]

9. Show the tree that you get if you insert 76 into tree $T_0$, using the basic algorithm that does not perform any rotations.

10. Show the tree that you get if you insert 76 into tree $T_0$, using the algorithm that performs rotations to keep the tree height-balanced. Start with the original tree $T_0$ that does not contain 76.
11. Show the tree that you get if you insert 97 into tree $T_0$, using the basic algorithm that does not perform any rotations. Start with the original tree $T_0$, which does not contain 76.

12. Show the tree that you get if you insert 97 into tree $T_0$, using the algorithm that performs rotations to keep the tree height-balanced. Start with the original tree $T_0$ that does not contain 76 or 97.
Question 13 refers to the following type definition, which defines a type of nodes in binary trees.

```cpp
struct Node
{
    int item;
    Node* left;
    Node* right;

    Node(int i, Node* L, Node* R)
    {
        item = i;
        left = L;
        right = R;
    }
};
```

13. Assume that tree \( t \) contains only positive integers. Write a C++ definition of function `largest(t)`, which returns the largest value in tree \( t \). If \( t \) is an empty tree, then `largest(t)` should return 0. Tree \( t \) is not necessarily ordered like a binary search tree. You can use the `max` function. A heading is given.

```cpp
int largest(const Node* t)
```