1. Find reasonable constants $c$ and $n_0$ to show that $n^2 + n \log_2(9n) = O(n^2)$. Show your work.

2. Solve the following recurrence, simplifying the result as much possible, and give its big-O.

   $$f(n) = \begin{cases} 
   1 & \text{if } n = 1 \\
   2f(n/2) + n/2 & \text{otherwise}
   \end{cases}$$

3. We have said in class that when something is $O(\log n)$, the actual base of the log doesn’t matter. Prove that this is true, namely that $\log_a n = O(\log_b n)$ for any choice of $a$ and $b$. *Hint: Try rewriting one logarithm in terms of the other.*
4. Suppose that you are writing a program to solve mazes. That is, you have a starting point, and you need to explore down each path, backtracking when you hit a dead end, until you get to a goal point. What data structure would be most suitable for keeping track of the path you’re currently following? Explain how you would use it.

5. Suppose you are given two dictionary implementations, one using a linked list and the other using a binary search tree. Which structure will allow for faster searches, and why?

6. Explain why a doubly-linked list would be a good way to implement both queues and stacks, compared to a growable array.
7. a) When deleting a node in a binary search tree, the removed node is replaced with the next smallest or next largest node in the tree. Why does this preserve the binary search tree property?

b) Here is Java class for binary search tree nodes.

class Node {
    String value;
    Node leftChild;
    Node rightChild;
    Node parent;
}

Write a Java function that takes a Node (the tree root) and a String value, and deletes that value from the tree.

c) What is the big-O of the deletion operation?