1. Provide solutions (using big-Oh or big-Theta) for the following recurrence relations.

   (a) \( T(n) = 3 \, T(n/2) + n \lg n \)
   (b) \( T(n) = 4 \, T(n/2) + n^2 \)
   (c) \( T(n) = 3 \, T(n/2) + n^2 \)

   \{ sol’n \}

   (a) Since \( \log_2 3 > 1 \), there is an \( \epsilon \) such that \( n \lg n = O(n^{\log_2 3 - \epsilon}) \). One such value would be \( \epsilon = (\log_2 3 - 1)/2 \). Thus \( T(n) = \Theta(n^{\log_2 3}) \).
   (b) Since \( \log_2 4 = 2 \) (and so \( n^2 = \Theta(n^{\log_2 4}) \)), we get \( T(n) = \Theta(n^2 \lg n) \).
   (c) Since \( \log_2 3 < 2 \), \( T(n) = \Theta(n^2) \).

2. Into an initially empty AVL tree, insert the following values:

   10, 30, 40, 25, 15, 5, 2, 42, 43, 44

   \{ sol’n \} For this and the next problem, please forgive my inability to typeset trees. I’m still working on it.

3. Insert the values above into an initially empty 2-3-4 tree. Then insert 45, 46, 47.

4. What is the run-time of the following pieces of code?

   (a) \( \text{for i = 1 to n \{ }
   \quad j = 1
   \quad \text{while (j<=i \{ }
   \quad \quad \text{sum++}
   \quad \quad j=2*j
   \quad \}\}
   \}

   \{ sol’n \}

   (a) The inner loop uses \( \lceil \lg i \rceil \) steps, so the whole thing takes \( \sum_{i=1}^{n} \lceil \lg i \rceil \) time. On one of the homeworks we saw that this is \( \Theta(n \lg n) \).
   (b) The inner loop was supposed to have been \( j=1 \) to \( i*i \). A typo gave you an easier problem, which is not too hard to see is \( \Theta(n^4) \).
5. Write a recursive routine which “flips” a binary tree - it should swap the left and right children of each node. The fields of each node are called key, lchild, rchild, and p (for parent).

```cpp
{ soln }

procedure flip( node cur) returns node
  if (cur == null) return null
  temp = flip(cur.lchild)
  cur.lchild = flip(cur.rchild)
  cur.rchild = temp
  return cur
```