Suppose are you given an instance sets of the MyDisjSets class representing a partition of
the set \{0, 1, \ldots, n - 1\} into disjoint subsets. Instances of MyDisjSets are stored as a single
array representing a collection of rooted trees as seen in class.

(a) What is the worst-case running time of labelEverything if we implement find without
path compression?

**Solution:** Each find operation touches at most \( n \) nodes in \( O(n) \) time. In total, labelEverything takes \( O(n^2) \) time.

On the other hand, consider the case where the tree is a path of \( n \) nodes. It takes \( \Omega(n) \)
time to perform a find from over half of those nodes, so labelEverything takes \( \Omega(n^2) \)
time total.

Together, the two arguments imply the worst-case running time to be \( \Theta(n^2) \).

**Rubric:** 4 points total. Solution must suggest some kind of \( \Omega(n^2) \) case exists for full credit.

(b) Prove that if we implement find using path compression, labelEverything runs in
\( O(n) \) time in the worst case.

**Solution:** We can think of each find operation as consisting of a (possibly empty) walk
up the tree to a grandchild of the root followed by some more operations that take constant
time. The time to walk to the grandchildren is proportional to the number of nodes on the
path. However, each node appears on the path to the grandchild at most once, implying
we spend \( O(n) \) time total doing the first halves of the find operations. We spend an
additional \( n \cdot O(1) = O(n) \) time doing the second halves, so labelEverything takes \( O(n) \)
time total.

**Rubric:** 6 points total.
Using $\Theta$-notation, provide asymptotically tight bounds in terms of $n$ for the solution to each of the following recurrences.

(a) $A(n) = 3A(n/3) + n$

**Solution:** Each level of the recursion tree sums to $n$, and there are $O(\log n)$ levels, so $A(n) = \Theta(n \log n)$. ■

(b) $B(n) = 2B(n/3) + n$

**Solution:** The $i$th level of the recursion tree sums to $(2/3)^i n$, so the sum of the level sums is a decreasing geometric series proportional to its first term, the value of the root. $B(n) = \Theta(n)$. ■

(c) $C(n) = 4C(n/3) + n$

**Solution:** The $i$th level of the recursion tree sums to $(4/3)^i n$, so the sum of level sums is an increasing geometric series proportional to its last term, the number of leaves. $C(n) = \Theta(n \log_3 4)$. ■

(d) $D(n) = 9D(n/3) + n^2$

**Solution:** Each level of the recursion tree sums to $n^2$, and there are $O(\log n)$ levels, so $D(n) = \Theta(n^2 \log n)$. ■

(e) $E(n) = 5E(n/2) + n^2$

**Solution:** The $i$th level of the recursion tree sums to $(5/4)^i n^2$, so the sum of level sums is an increasing geometric series proportional to its last term, the number of leaves. $E(n) = \Theta(n \log_2 5)$. ■

**Rubric:** 10 points total: 2 points per part, all or nothing. No proofs are needed for full credit.