CS330 Recitation 11*

Part 1: Casting out nines

- There’s a method of verifying arithmetic computations (+, -, *, /) on positive natural numbers called “casting out nines”.
- The digit sum of an integer ≥ 1 is the value you get when you add together the digits of the integer. E.g., the digit root of 634 = 6 + 3 + 4 = 16.
- To get the digit root of an integer ≥ 1, you calculate its digit sum; if the sum is ≤ 9, then that’s the digit root; if not, you calculate the digit sum of the digit sum, etc. Repeat until you get a value ≤ 9. E.g., for 634, we calculate 6 + 3 + 4 = 16, then 1+6 = 7, so 7 is the digit root of 634.
- Exception: If you get a sum of 0, you use 9 as the digit root instead.

Questions
1. (a) Calculate the digit root \( r \) of 34. (b) What is the difference \( D = 34 - r \) and is it divisible by 9?

2. Consider the two-digit number \( N = d_1d_0 \). (We’re concatenating two digits that are each between 0 and 9, so e.g., 34 corresponds to \( d_1 = 3 \) and \( d_0 = 4 \).)
   a. Argue that \( N \equiv \) its digit root \( \text{mod} \ 9 \). (There are two cases; one where \( d_1 + d_0 \leq 9 \) and one where it’s > 9.)
   b. Prove that any positive integer \( N \) is \( \equiv \) its digit root \( \text{mod} \ 9 \). (The base cases are 1 and 2 digit numbers. For more digits, generalize from the observation that the digit root of 634 = the digit root of (the digit root of 600 + the digit root of 34) = the digit root of 67.

3. Define +, -, *, and / on digit roots to be regular +, -, *, / except that you take the digit root of the result. Let \( R(\ldots) \) map numbers to their digit roots. Argue that \( R(\ldots) \) “respects +, -, *, /” in the following sense: For all integer \( M \) and \( N > 0 \), \( R(M + N) = R(M) + R(N) \) and similarly for -, *, and /. (Or come up with counterexamples.)

4. So now we know \( R(S) = R(M) + R(N) \) is a necessary condition for \( S = M + N \). Is it a sufficient condition? (Either argue that it is or come up with a counterexample.)

Part 2

5. One technique for writing a spellchecker is to use a hash table of size ≥ the number of words you want to know how to spell: Initialize the table so that every entry holds \texttt{false} and then for each word \( W \) that you want to include in the spellchecker, set the hash table entry \( \text{hash}(W) \) to \texttt{true}. Briefly discuss the pros and cons of this approach.

6. Create a division table modulo 11. (The entry for row \( a \) column \( b \) is \( c \) where \( a \equiv b \cdot c \ \text{mod} \ 11 \).

---

* Some material from MATH230 Dr. Robert Ellis IIT-AMAT, zyBooks Disc. Math., and Disc. Math. and its Appl., Rosen (7th Ed.) is included under the academic/nonprofit Fair Use exception of US Copyright Law. This document must NOT be distributed beyond participants in IIT’s CS330.