CS 6301: Implementation of data structures and algorithms
Short Project 3: Integer arithmetic with arbitrarily large numbers

Ver 1.0: Initial description (Fri, Feb 3).
Due: 11:59 PM, Fri, Feb 10.

- Submission procedure is same as the same as that of prior projects.
- For each group, only the last submission is considered. Earlier submissions are discarded.
- Your code must be of good quality, and pass all test cases to earn full credit.

Project Description

In this project, develop a program that implements arithmetic with large integers, of arbitrary size.

Do not use BigInteger, BigNum, or other libraries that implement arbitrary precision arithmetic.

The starter code Num.java is in shared class folder. Testcases will also be provided

Your task is to implement the class Num that stores and performs arithmetic operations on arbitrarily large integers. You must use the following data structure for representing Num: Array of long integers, where the digits are in the chosen base. Particularly, do not use strings to represent the numbers. Each entry of the list stores exactly one long integer. Assume the base to 10.

Implement the following methods. Since subtract method is optional assume the input numbers are positive.

- Num(String s): Constructor for Num class; takes a string s as parameter, with a number in decimal, and creates the Num object representing that number in the chosen base. **Note that, the string s is in base 10.** The string s can have arbitrary length.
- String toString(): convert the Num class object into its equivalent string (in decimal). There should be no leading zeroes in the string.
- Num add(Num a, Num b): sum of two numbers a+b stored as Num.
- Num product(Num a, Num b): product of two numbers a*b.
- printList(): Print the base + ":" + elements of the list, separated by spaces.

The following methods are optional.

- Num subtract(Num a, Num b): a-b
- Num power(Num x, long n): given an Num x, and n, returns the Num corresponding to x^n (x to the power n). Assume that n is a nonnegative number. Use divide-and-conquer to implement power using O(log n) calls to product and add.

- Num divide(Num a, Num b): Integer division a/b. Use divide-and-conquer or division algorithm. Return null if b=0.

Project

The input is a string of arbitrary length. A trivial implementation will store each digit in an element of array. For example, if 4832 is input, you can parse the string and store it as follows.

list[0] is 2, list[1] is 3, list[2] is 8, and list[3] is 4.

Similarly, 5167 will be stored as:

list[0] is 5, list[1] is 1, list[2] is 6, and list[3] is 7.

Then you will add the corresponding elements of the two arrays and store the result.

list[0] is 9, list[1] is 9, list[2] is 9, and list[3] is 9.

A better implementation is as follows.

list[0] is 32, and list[1] is 48,

The above is possible because list is an array of long.

Similarly, 5167 will be stored as:

list[0] is 51, and list[1] is 67.

Then you will add the corresponding elements of the two arrays and store the result.

list[0] is 99, and list[1] is 99.

The number additions is cut down to 2 in this case. But your code can be more efficient than this. The most efficient implementation will store as large a value possible in each element of the array list. Figure it out.

You may need to extract substrings from the input string and convert each substring into a long integer. Look for suitable methods in String and Long classes.