CS 3353 Fall 2024

Programming Homework 3 – Big Three

Due: 11/26 (Tue) 11:59pm

You are to define a class called BigThree, which stores numbers of base 3 of arbitrary length.

Inside your class, you are to represent the number by a string, which store the digits. How you are going to arrange the digits is totally up to you.

You should implement the following methods for the class:

- Constructor:
  - BigThree(): initialize the string to '0'
  - BigThree(string s): first check whether the string contains only digits of 0, 1, 2. If so, initialize the number to the string, otherwise, initialize the string to '0'.
  - BigThree(BIgThree bt): just copy the value of bt to the new object.
- Methods:
  - BigThree Add(BigThree s1): Add s1 to the current object and return the result as a new BigThreeObject.
  - BigThree Subtract(BigThree s1): subtract s1 to the current number and return the result as a new Three. If s1 is larger than the current value, then return 0.
  - BigThree Shift(BigThree s1, int k): return a BigThree by adding k zeros at the end of s1. (if k <= 0, the return object should have the same value of s1).
  - BigThree MultiplyDivideAndConquer(const BigThree& s1): multiply s1 to the current number and return the result as a new BigThree. You should apply the divide and conquer method we mentioned in class (video). Your running time should be O($n^{\log_2 3}$), if the length of both strings are n.
  - string Str() const: return a string for the digits of the BigThree, in the order as it should be displayed.

In your implementation, you are NOT allowed to use the addition and multiplication operator (nor should you just write a loop to add numbers multiple times) to manipulate the digits. You can still use the operator in a for() statement. Instead, you are given methods (as methods of the BigThree class).

- char add_base(char x, char y): x and y should be single character of 0, 1, 2, and are treated as digit (base 3). The method returns the rightmost digit of the sum (x+y) (base 3).
- boolean add_carry(char x char y) x and y should be single character of 0, 1, 2, and are treated as digit (base 3). The method returns true if adding them (x+y) results in a carry
- char sub_base(char x, char y): x and y should be single character of 0, 1, 2, and are treated as digit (base 3). The method returns the rightmost digit of the difference (x-y) (base 3).
- Boolean sub_borrow(char x char y) x and y should be single character of 0, 1, 2, and are treated as digit (base 3). The method returns true if x-y returns in a borrow
- char multi_base(char x, char y): x and y should be single character of 0, 1, 2, and are treated as digit (base 3). The method returns the rightmost digit of the sum (x*y) (base 3).
- Boolean multi_carry(char x char y) x and y should be single character of 0, 1, 2, and are treated as digit (base 3). The method returns true if adding them (x*y) results in a carry

You should always call the multiply_base method for multiplication and add_base for addition for individual digits, and sub_base for subtraction of individual digits.

Notice you are NOT allowed to convert the string into numbers and do the calculation and convert them back into strings. Doing that will means you will get 0 point for the program.

What to hand in

You will be given a file call BigThree.java that contains declarations and some code for the class, you are to implement the rest of the method and upload the .java file.

You should not implement a main() method in the class. You are welcomed to write your own driver program to test the results. A sample will be given to you to do some testing.

Extra Credit

You should write the following methods for

- (25 points) BigThree Power10 (int k): convert the (decimal) number 10^k into a base four number and return it as a BigThree. You need to use divide and conquer to do it.
    - For base cases (k=0, 1), just simply hard-code the answer and return.
    - Otherwise, divide the number to two parts, recursively call Power10_2, and combine them using the appropriate method(s) that you implemented above

- (25 points) BigThree FromDecimal(string s): s is a numeric string storing a decimal integer. The method should return a BigThree corresponding to s, using the following divide and conquer method
    - Base case is s has only one digit. Just hard-code all 10 possibility and return.
    - Otherwise, divide the string into two halves, and use recursion and method(s) you implemented above return the answer. (Hint: 3483 = 34 * 100 + 83)

For your reference. Here is what will be returned for the methods above:

X	Y	Add_base	Add_carry	Sub_Base	Sub_borrow	Multi_base	Multi_carray
'0'	'0'	'0'	False	'0'	False	'0'	False
'0'	'1'	'1'	False	'2'	True	'0'	False
'0'	'2'	'2'	False	'1'	True	'0'	False
'1'	'0'	'1'	False	'1'	False	'0'	False
'1'	'1'	'2'	False	'0'	False	'1'	False
'1'	'2'	'0'	True	'2'	True	'2'	False
'2'	'0'	'2'	False	'2'	False	'0'	False
'2'	'1'	'0'	True	'1'	False	'2'	False
'2'	'2'	'1'	True	'0'	False	'1'	True