

Project – 5

Kumaran Rajendiran (kxr131430)

Arvind Krishna Parthasarathy (axp133230)

Problem Description:

Perform arithmetic operations such as addition, subtraction, multiplication, division, computing power and square root, on linked lists representing numbers.

Methodology:

All operations are done in BASE 10.

1. **strToNum** : A string of numbers is the input here and it outputs a linked list of integers with each character of the string as a single node in the linked list.
"1234" to 4 -> 3 -> 2 -> 1.
2. **numToStr**: A linked list of integers is given as input and it outputs a String representing the number.
1 -> 2 -> 3 -> 4 to "4321"
3. **Addition**: The addition of two linked lists is similar to addition of integers. We start at the head of the two linked lists and iterate synchronously through them while calculating the sum and carry at each step.
4. **Subtraction**: Subtraction is done by the conventional method too. If the value at the first list iterator is lesser than the second list iterator value, we update carry to 1 which means we are borrowing from the next node.

We also handled the negative result case. When we run out of values at both the linked lists, and the carry is one, we know that the first number is smaller than the second. Here we subtract the result from $10^{(\text{length of result}+1)}$ and add a negative sign before the number.

5. **Multiplication**: Done using Karatsuba algorithm for multiplication. References – Wiki
http://en.wikipedia.org/wiki/Karatsuba_algorithm#Implementation
6. **Power**: Uses karatsuba algorithm for multiplication, while exponentially decreasing the power in each step until it becomes zero.
7. **Division and Modulus**: Suppose we have $654 / 4$. We multiply 4 by 100 to make it's size equal to that of the dividend. We then add the same number (400) to it to see if it exceeds the actual

number, if it does, we move on to the next digit. We do this until we run out of digits. Quotient is updated then and there. Returns quotient or remainder based on the Boolean parameter.

8. **Square root:** Similar to division. Here we take the last half of the list representing the number. If the size of the list is odd, we add 1 to the beginning of the list. We use the same procedure as in division but here we compute square and find if the computed value is more than the actual value. If it is greater, we go back by one step and then move on to the next digit. We do this until we run out of digits.
9. **Power in 15 seconds:** To return the maximum power of a number that can be computed in 15 seconds. Suppose the given number is represented by a list a . We start with a^2 , then $(a^2)^2$ and so on until we run out of time. When we run out of time, we discard the last value that was calculated and return the previous value because it was not calculated in time.

Development Platform:

IDE used: Eclipse

OS: Windows 8

Hardware: 8GB RAM, 64 bit OS, Intel i7 @ 2.40GHz

Test results:

For the sample input files on your website:

In-1.txt:

Running time (msec):

16537 including the 15 seconds for power.

In-2.txt:

Running time (msec):

176459

In-3.txt:

Running time (msec):

8657

In-4.txt:

Running time (msec):

88768

Time taken to complete the project:

2 days.

References:

Karatsuba algorithm for multiplication – Wikipedia.

http://en.wikipedia.org/wiki/Karatsuba_algorithm