

Large number multiplication

Introduction

For this assignment I compared two different multiplication algorithms and tested to see how much faster the Karatsuba algorithm is from the standard multiplication algorithm that is taught to most students in schools.

Algorithms

Below is the Karatsuba Algorithm that runs at a time complexity of $n^{\log_2(3)}$

```
def karatsuba_algorithm(x,y):
    if x < 10 or y < 10:
        return x*y

    n = max(int(log10(x)+1) , int(log10(y)+1))
    n_2 = int(ceil(n/2.0))

    if n % 2 == 1:
        n +=1

    a,b = divmod(x, 10**n_2)
    c,d = divmod(y, 10**n_2)

    ac = karatsuba_algorithm(a,c)
    bd = karatsuba_algorithm(b,d)

    ad_bc = karatsuba_algorithm((a+b) , (c+d)) - ac - bd
    return (((10**n)*ac) + bd + ((10**n_2) * (ad_bc)))
```

Below is the Standard Algorithm that runs at a time complexity of n^2

```
def standard_algorithm(num1, num2):
    str_num1 = str(num1)
    str_num2 = str(num2)

    str_num1 = str_num1[::-1]
    str_num2 = str_num2[::-1]

    result = 0

    for i in range(len(str_num1)):
        carry = 0

        for j in range(len(str_num2)):
            prod = int(str_num1[i]) * int(str_num2[j]) + carry

            if i+j == 0:
                result = prod
            else:
                prev_result = (result // 10**(i+j)) % 10
                result -= prev_result * 10**(i+j)
                result += (prev_result + prod % 10) * 10**(i+j)
                carry = prod // 10

        if carry > 0:
            result += carry * 10**(i+len(str_num2))

    return result
```

Results

Below are the results of running three tests on two algorithms speeds, the numbers they were multiplying were 3141592653589793238462643383279502884197169399375105820974944592 and 2718281828459045235360287471352662497757247093699959574966967627. They both got the result of

8539734222673567065463550869546574495034888535765114961879601127067743044893204848617875072216249073013374895871952806582723184

As is shown by the results the Karatsuba algorithm is faster than the standard version by about 7-8 times. If your program works with large amounts of extremely large numbers such as in the case of cryptography, then using an algorithm like Karatsuba would be useful.

```
Standard: 0.0080731 seconds
Karatsuba: 0.0010690999999999895 seconds
Karatsuba algorithm is faster by: 0.00700400000000001 seconds
```

```
Standard: 0.009742400000000005 seconds
Karatsuba: 0.001242000000000007 seconds
Karatsuba algorithm is faster by: 0.008500399999999998 seconds
```

```
Standard: 0.009469899999999996 seconds
Karatsuba: 0.0013058000000000028 seconds
Karatsuba algorithm is faster by: 0.008164099999999994 seconds
```