COMP-250 Homework One

Neil Edelman - 110121860

2007-09-16

These graphs take a slice of the $\mathbb{R}_3(a, b, t)$ space defined by $a = b$, which should be the worst-case. See Figures 5 - 7 in the Appendix.

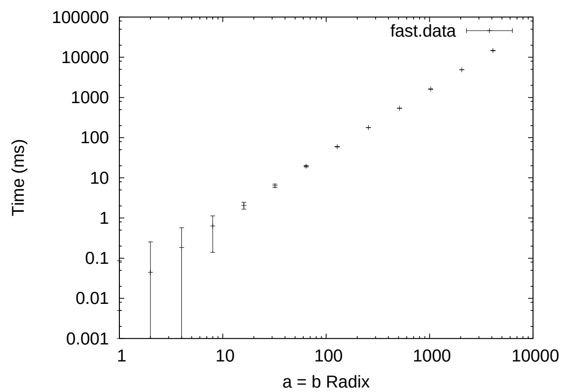

Figure 1: Multiplication by Karatsuba Recusion Algorithm.

In Figure 1, it's showing a small, but definate trend downwards. I think.

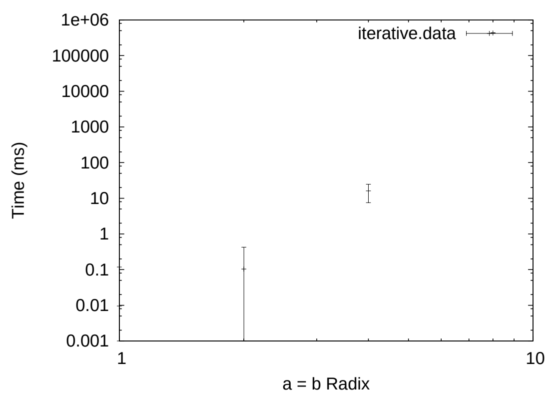

Figure 2: Multiplication by iterating addition.

In Figure 2, it blows up. We should find that the varience is huge because the function is dependent on the content of the numbers, not just the digits. It's too small a sample to tell.

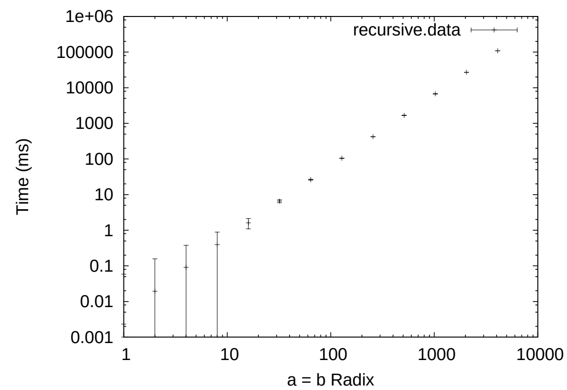

Figure 3: Multiplication by recursion.

In Figure 3, it looks pretty stright. Likely depends on the square.

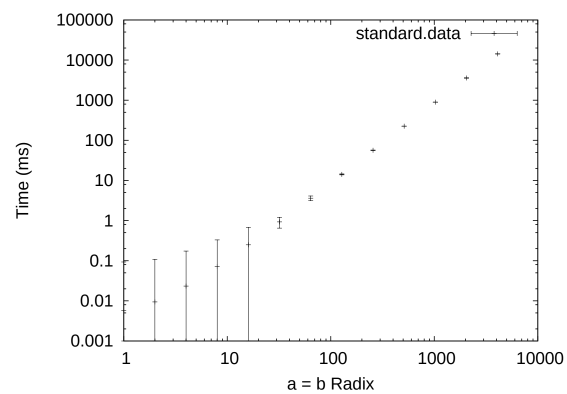

Figure 4: Multiplication by the Standard Long Algorithm.

In Figure 4, it looks pretty stright; it likely depends on the square as well.

1 Question 4

a) See Table 1 for results.

digits	Karatsuba	Iterative	Recursive	Long
1	0.005±0.08	0.01±0.1	0.002±0.06	0.006±0.09
2	0.04±0.2	0.1±0.3	0.02±0.1	0.009±0.1
4	0.2±0.4	16±9	0.09±0.3	0.02±0.2
8	0.6±0.5	429395	0.4±0.5	0.07±0.3
16	2.1±0.4		1.6±0.5	0.2±0.4
32	6.4±0.6		6.6±0.6	0.9±0.3
64	19.5±0.9		26.2±0.7	3.6±0.5
128	59.3±0.8		105.1±0.9	14.2±0.4
256	178.8±0.7		422.7±0.9	56.4±0.7
512	538.5±0.5		1672	225.2±0.7
1024	1622		6738	894.5±0.5
2048	4888		27146	3577
4096	14676		108944	14289
projected 8192	40000	4×10^{8192}	400000	60000
projected 16384	100000	9×10^{16384}	2000000	200000
n	$0.028 \cdot n^{\log_2 3}$	$0.00054 \cdot 10^{n+1}$	$0.0065 \cdot n^2$	$0.00085 \cdot n^2$

Table 1: This shows t_{mean} (ms) for the experement. Numbers where no error is given had one replica.

b,c,d) I used the maximum time to calculate the constant, then I did a regression check to verify the analysis. The Karatsuba Algorithm is $O(n^{\log_2 3})$. I Wikipediad that shit.

$$\begin{aligned} n &= \frac{14676}{4096^{\log_2 3}} \\ n \times 128^{\log_2 3} &= 60 \\ n \times 8192^{\log_2 3} &= 40000 \\ n \times 16384^{\log_2 3} &= 100000 \\ t &= 0.028 \cdot n^{\log_2 3} \end{aligned}$$

The Iterative Algorithm depends on the contents of the number. Assuming addition is linear (see Figure 5.)

$$\begin{aligned} n &= \frac{429395}{8 \cdot 10^8} \\ n \times 8192 \cdot 10^{8192} &= 4 \times 10^{8192} \\ n \times 16384 \cdot 10^{16384} &= 9 \times 10^{16384} \\ t &= 0.00054 \cdot 10^{n+1} \end{aligned}$$

The Recursive Algorithm was found to depend on the square by regression.

$$
\begin{aligned}
n &= \frac{108944}{4096^2} \\
n \times 128^2 &= 106 \\
n \times 8192^2 &= 400000 \\
n \times 16384^2 &= 2000000 \\
t &= 0.0065 \cdot n^2
\end{aligned}
$$

The Long Multiplcation Algorithm should was also found to scale with the square.

$$
\begin{aligned}
n &= \frac{14289}{4096^2} \\
n \times 128^2 &= 14 \\
n \times 8192^2 &= 60000 \\
n \times 16384^2 &= 200000 \\
t &= 0.00085 \cdot n^2
\end{aligned}
$$

e) We compare Long Multiplication. The analysis for the others is similar.

$$
\begin{aligned}
0.0065 \cdot n^2 &= 0.028 \cdot n^{\log_2 3} \\
n^{0.42} &= 32.94 \\
n &= e^{\frac{\ln 32.94}{0.42}} \\
n &= 4000
\end{aligned}
$$

Well, we noticed that around four-thousand it was catching up so it seems reasonable. This is just a rough estimate.
f) When it's 20^{20}?
Karatsuba

$$
\begin{aligned}
t &= 0.028 \cdot 20^{20 \log_2 3} \\
&= 4.9 \times 10^{39}
\end{aligned}
$$

Iterative

$$
\frac{0.00054 \cdot 10^{20^{20}+1}}{4.9 \times 10^{39}} = 1.1 \cdot 10^{104857599999999999999958}
$$

Recursive

$$
\frac{0.0065 \cdot 20^40}{4.9 \times 10^{39}} = 10000000000
$$

Long

$$
\frac{0.00085 \cdot 20^40}{4.9 \times 10^{39}} = 2000000000
$$

A Appendix

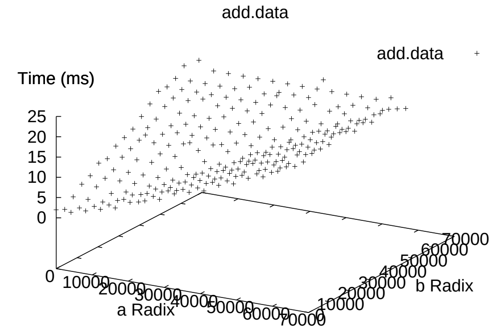

Figure 5: Addition of two random numbers a = b.

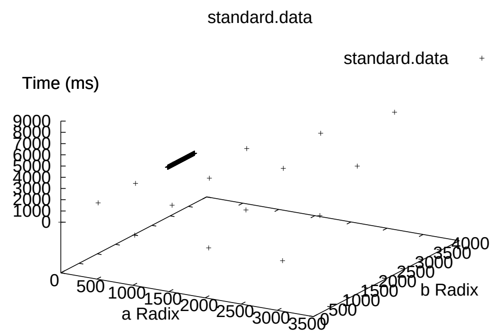

Figure 6: Standard Multipication of two numbers.

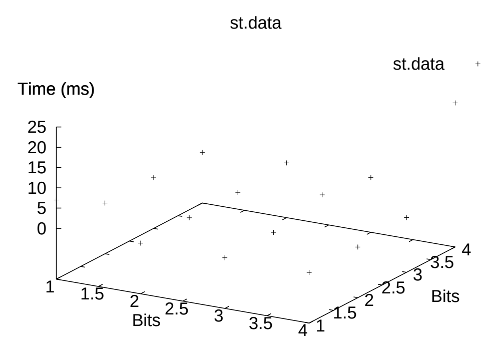

Figure 7: Standard Multipication of two numbers.