Today's topics

- Algorithms
- Complexity
- Upcoming
  - AI
- Reading
  - *Brookshear 5.6*

New machines vs. new algorithms

- New machine.
  - Costs $$$ or more.
  - Makes "everything" finish sooner.
  - Incremental quantitative improvements (Moore's Law).
  - May not help much with some problems.
- New algorithm.
  - Costs $ or less.
  - Dramatic qualitative improvements possible! (million times faster)
  - May make the difference, allowing specific problem to be solved.
  - May not help much with some problems.
- Algorithmic Successes
  - N-body Simulation, Discrete Fourier transform, Quantum mechanical simulations, Pixar movies…

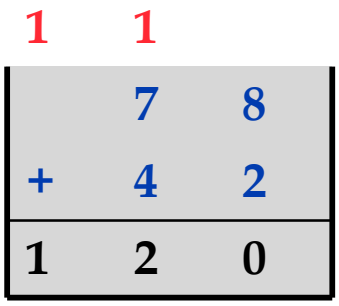
Algorithms

- What is an algorithm?

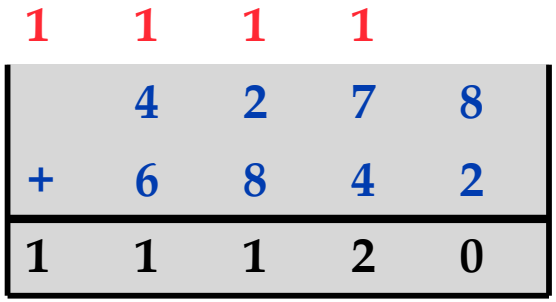
- So far we have been expressing our algorithms in Java code
- *Pseudocode* is a more informal notational system
  - Can't be too pseudo. Should still be able to derive real code.
  - Worry about the problem solving and not compilation errors, file permission, or browser settings
- Coming up with solution is just the first problem
- For many problems, there may be several competing algorithms
- Computational complexity
  - Rigorous and useful framework for comparing algorithms and predicting performance

Linear Growth

- Grade school addition
  - Work is proportional to number of digits N
  - *Linear* growth: kN for some constant k



N = 2

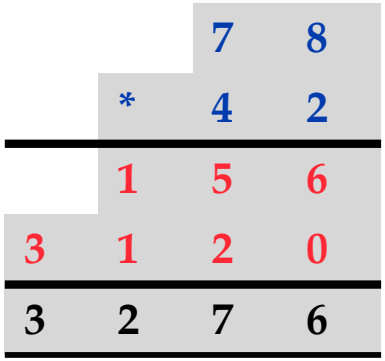


N = 4

- How many reads? How many writes? How many operations?

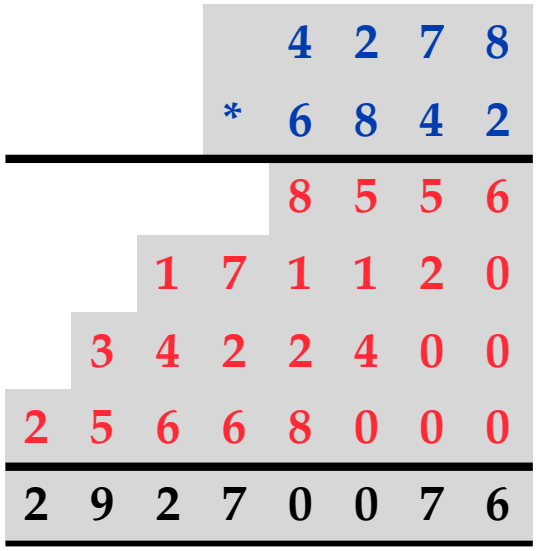
Quadratic Growth

- Grade school multiplication
  - Work is proportional to *square* of number of digits N
  - *Quadratic* growth: kN^2 for some constant k



		7	8
	*	4	2
	1	5	6
3	1	2	0
3	2	7	6

N = 2



			4	2	7	8	
		*	6	8	4	2	
			8	5	5	6	
		1	7	1	1	2	0
	3	4	2	2	4	0	0
2	5	6	6	8	0	0	0
2	9	2	7	0	0	7	6

N = 4

- How many reads? How many writes? How many operations?

Sorting

- Given n items, rearrange them so that they are in increasing order
- A key recurring problem
- Many different methods, how do we choose?
- Given a set of cards, describe how you would sort them:

- Given a set of words, describe how you would sort them in alphabetical order?

Why Does It Matter?

Run time (nanoseconds)		$1.3\ N^3$	$10\ N^2$	$47\ N \log_2 N$	$48\ N$
Time to solve a problem of size	1000	1.3 seconds	10 msec	0.4 msec	0.048 msec
	10,000	22 minutes	1 second	6 msec	0.48 msec
	100,000	15 days	1.7 minutes	78 msec	4.8 msec
	million	41 years	2.8 hours	0.94 seconds	48 msec
	10 million	41 millennia	1.7 weeks	11 seconds	0.48 seconds
Max size problem solved in one	second	920	10,000	1 million	21 million
	minute	3,600	77,000	49 million	1.3 billion
	hour	14,000	600,000	2.4 trillion	76 trillion
	day	41,000	2.9 million	50 trillion	1,800 trillion
N multiplied by 10, time multiplied by		1,000	100	10+	10

Orders of Magnitude

Seconds	Equivalent
1	1 second
10	10 seconds
10^2	1.7 minutes
10^3	17 minutes
10^4	2.8 hours
10^5	1.1 days
10^6	1.6 weeks
10^7	3.8 months
10^8	3.1 years
10^9	3.1 decades
10^{10}	3.1 centuries
. . .	forever
10^{21}	age of universe

Meters Per Second	Imperial Units	Example
10^{-10}	1.2 in / decade	Continental drift
10^{-8}	1 ft / year	Hair growing
10^{-6}	3.4 in / day	Glacier
10^{-4}	1.2 ft / hour	Gastro-intestinal tract
10^{-2}	2 ft / minute	Ant
1	2.2 mi / hour	Human walk
10^2	220 mi / hour	Propeller airplane
10^4	370 mi / min	Space shuttle
10^6	620 mi / sec	Earth in galactic orbit
10^8	62,000 mi / sec	1/3 speed of light

Powers of 2		
	2^{10}	thousand
	2^{20}	million
	2^{30}	billion