Lecture 1

Grade school algorithms for arithmetic

Fri. Sept. 7, 2018

What is an algorithm?

An algorithm is a sequence of instructions or operations for manipulating data to produce some result. (Algorithms existed before computers!)

Think of an algorithm as a recipe. In CS, the recipe works with digital information such as numbers, text strings, images, sounds,….

See Khan Academy course on Algorithms for a good intro

See also this wonderful Netflix series on Algorithms.

Today: grade school arithmetic

The first algorithms you ever learned:

- addition
- subtraction
- multiplication
- division

First steps ....

How did you learn arithmetic when you were a child?

First you (maybe) learned to count on your fingers.

Second, you memorized the sum of single digit numbers:

1+3 = 5
4+7= 11, etc

Then, you learned to add single digit sums.   How?

Grade school addition

$$
\begin{array}{rcccc}
 & 2 & 3 & 4 & 9 \\
+ & 5 & 8 & 1 & 3 \\
\hline
\end{array}
$$

Grade school addition

$$
\begin{array}{ccccccc}
 & \color{red}{1} & \color{red}{0} & \color{red}{1} & & \color{red}{\longleftarrow} & \color{red}{\text{carry}} \\
 & 2 & 3 & 4 & 9 & & \\
+ & 5 & 8 & 1 & 3 & & \\
\hline
 & 8 & 1 & 6 & 2 & &
\end{array}
$$

You needed to memorize single digit sums to do this.

What is the algorithm for addition $a + b$?

Let's use an array for a, b, and the result r.

$$
\begin{array}{rccccc}
 & & a[3] & a[2] & a[1] & a[0] \\
 & + & b[3] & b[2] & b[1] & b[0] \\
\hline
 & r[4] & r[3] & r[2] & r[1] & r[0]
\end{array}
$$

Grade School Addition

For each column i {

compute single digit sum a[i] + b[i] and
add the "carry value" from previous column

determine the single digit result r[i] for that column
and the carry value for the next column

}

Grade School Addition ("pseudocode")

$$
\begin{array}{l}
carry = 0 \\
\textbf{for } i = 0 \text{ to } N-1 \ \ \textbf{do} \\
\quad r[i] \leftarrow (a[i] + b[i] + carry) \ \% \ 10 \\
\quad carry \leftarrow (a[i] + b[i] + carry)/10 \\
\textbf{end for} \\
r[N] \leftarrow carry
\end{array}
$$

(To be explained on next slides.)

Grade School Addition ("pseudocode")

compute single digit sum a[i] + b[i] and
add the carry value from previous column
determine the result r[i] for that column

"mod"

$$
\begin{aligned}
&carry = 0 \\
&\textbf{for } i = 0 \text{ to } N-1 \textbf{ do} \\
&\quad \boxed{r[i] \leftarrow (a[i] + b[i] + carry) \% 10} \\
&\quad carry \leftarrow (a[i] + b[i] + carry)/10 \\
&\textbf{end for} \\
&r[N] \leftarrow carry
\end{aligned}
$$

Grade School Addition ("pseudocode")

$carry = 0$

for $i = 0$ to $N-1$ **do**

$\quad r[i] \leftarrow (a[i] + b[i] + carry)\ \%\ 10$

$\quad \boxed{carry \leftarrow (a[i] + b[i] + carry)/10}$

end for

$r[N] \leftarrow carry$

Integer division
(ignore remaider)

determine the carry value for the next column

The grade school addition algorithm is non-trivial.

It makes use of a good *number representation*, namely it represents each number as *sum of powers of 10*.

(Hindu-Arabic system invented ~2000 years ago)

Imagine an algorithm for addition that is based on Roman numerals:

It would be rather awkward!

Roman Numeral Table							
1	I	14	XIV	27	XXVII	150	CL
2	II	15	XV	28	XXVIII	200	CC
3	III	16	XVI	29	XXIX	300	CCC
4	IV	17	XVII	30	XXX	400	CD
5	V	18	XVIII	31	XXXI	500	D
6	VI	19	XIX	40	XL	600	DC
7	VII	20	XX	50	L	700	DCC
8	VIII	21	XXI	60	LX	800	DCCC
9	IX	22	XXII	70	LXX	900	CM
10	X	23	XXIII	80	LXXX	1000	M
11	XI	24	XXIV	90	XC	1600	MDC
12	XII	25	XXV	100	C	1700	MDCC
13	XIII	26	XXVI	101	CI	1900	MCM

Grade school subtraction

$$\begin{array}{r} 924 \\ -\ \underline{352} \\ 572 \end{array}$$

How to write an algorithm for doing this?
You *could* do it.   (Assignment 1 in previous years.)

Grade school subtraction

$$\begin{array}{r} {\color{red}8}\;\;{\color{red}{}^{1}} \\ \cancel{9}24 \\ -\;\;\underline{352} \\ 572 \end{array}$$

How to describe the "borrowing" step?

Multiplication

Q: What do we mean by a * b ?

(assuming integers)

Multiplication

Q: What do we mean by $a * b$?

A: We mean: $(a + a + ... + a)$, b times

a is the "multiplicand"

b is the "multiplier"

Multiplication

Q: What do we mean by $a * b$?

A: (a + a + ….. + a), b times

or (b + b + … + b), a times

Slow Multiplication Algorithm

$$
\begin{array}{l}
product = 0 \\
\textbf{for } i = 1 \text{ to } b \ \ \textbf{do} \\
\quad product \leftarrow product + a \\
\textbf{end for}
\end{array}
$$

You learned a *much* faster algorithm in grade school.

Grade school multiplication

Example:

352	a[N]	“multiplicand”
* 964	b[N]	“multiplier”

Grade school multiplication

Example:

352	a[N]	"multiplicand"
* 964	b[N]	"multiplier"

Algorithm:

For each digit in b[ ]
  for each digit in a[ ]
    multiply the two digits and do what ?...

Grade school multiplication
(Example)

Step 1: make 2D table tmp [ ][ ]

$$
\begin{array}{rl}
352 & \quad a[\mathrm{N}] \\
*\ \underline{964} & \quad b[\mathrm{N}]
\end{array}
$$

2

1408

3 1

21120 $\qquad tmp[2\mathrm{N}][\mathrm{N}]$

4 1

<u>316800</u>

Grade school multiplication
(“pseudocode”)

Step 1: make 2D table tmp[ ][ ]

$\textbf{for } j = 0 \text{ to } N-1 \textbf{ do}$
$\quad carry \leftarrow 0$
$\quad \textbf{for } i = 0 \text{ to } N-1 \textbf{ do}$
$\quad\quad prod \leftarrow (a[i] * b[j] + carry)$
$\quad\quad tmp[j][i+j] \leftarrow prod\%10$
$\quad\quad carry \leftarrow prod/10$
$\quad \textbf{end for}$
$\quad tmp[j][N+j] \leftarrow carry$
$\textbf{end for}$

Grade school multiplication
(Example)

Step 2: add the columns of the 2D table

352	a[N]
* 964	b[N]
1	
1408	
21120	tmp[2N][N]
316800	
339328	r[2N]

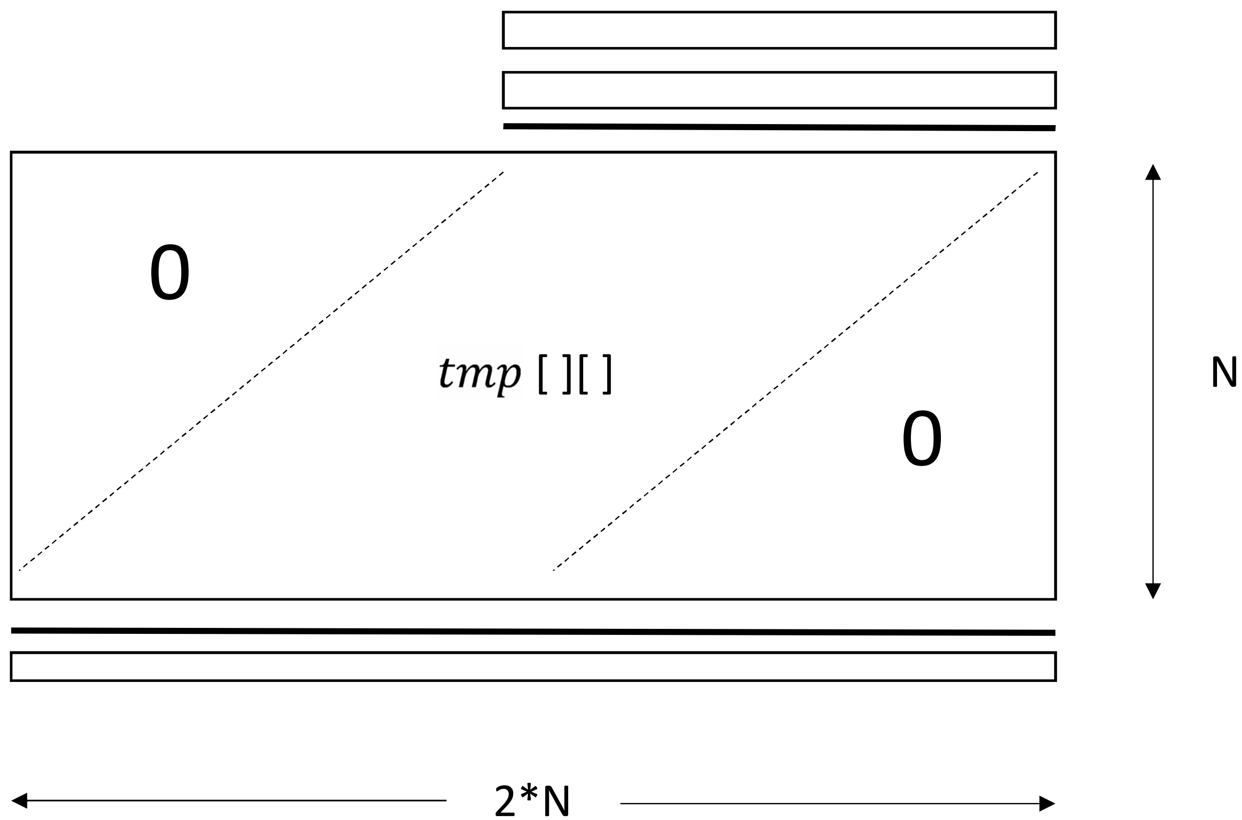
Grade school multiplication
(Pseudocode)

Step 2: for each column in table, sum up the single digits in the rows and add the carry

$carry \leftarrow 0$
for $i = 0$ to $2 * N - 1$ **do** // columns
  $sum \leftarrow carry$
  for $j = 0$ to $N - 1$ **do** // rows
    $sum \leftarrow sum + tmp[j][i]$
  end for
  $r[i] \leftarrow sum\%10$
  $carry \leftarrow sum/10$
end for

ASIDE: Grade school multiplication specifies that we build a temporary 2D array of size 2*N*N.

Q: Is a 2D tmp [ ][ ] array necessary?



ASIDE: Grade school multiplication specifies that we build a temporary 2D array of size 2*N*N.

Q: Is a 2D tmp [ ][ ] array necessary?

A: No.

(We could instead just add each row to a running sum as we go. We still need to compute all the rows, but we don't need to compute them all in advance.)

Division

Q: What do we mean by a/b ?

(Assume they are integers, and $a > b$.)

Division

Q: What do we mean by a/b ?
(Assume they are integers, and $a > b$)

A: We mean: "How many times can we subtract b from a before our answer is between 0 and b ?"

Division

Q: What do we mean by a/b ?

(Assume they are integers, and $a > b$)

A: $a = q * b + r, \qquad 0 \le r < b$

q is quotient, r is remainder

Slow division algorithm

To compute a / b, repeatedly subtract b from a until the result is less than b.

$$
\begin{array}{l}
q = 0 \\
r = a \\
\textbf{while } r \geq b \ \ \textbf{do} \\
\quad q \leftarrow q + 1 \\
\quad r \leftarrow r - b \\
\textbf{end while}
\end{array}
$$

You learned a much faster algorithm in grade school.

Grade school division ("long division")

```
            5 ...
       _______________
723 | 41672542996
      3615
        ----
         552  ...etc
```

It is not so easy to write this as an algorithm.
You *could* do it.   (Assignment 1 in previous years.)

Computational Complexity

What do we mean by 'fast' and 'slow'?

Let $t(N)$ be the number of "steps" of a computation whose "input size is N".

e.g. In arithmetic, we assume the numbers we are operating on each have N digits.

How many steps are required for addition?

How many steps are required for multiplication ?

Grade School Addition

$$
\begin{array}{ll}
carry = 0 & 1 \\
\textbf{for } i = 0 \text{ to } N-1 \ \ \textbf{do} & \\
\left.\begin{array}{l}
\quad r[i] \leftarrow (a[i] + b[i] + carry) \ \% \ 10 \\
\quad carry \leftarrow (a[i] + b[i] + carry)/10
\end{array}\right\} & N \\
\textbf{end for} & \\
r[N] \leftarrow carry & 1
\end{array}
$$

We mean that each part of the program is executed 1 or N times.

As you will learn in COMP 273, some operations take more time than others.

How could we express this?

Grade School Addition

$carry = 0$ $\quad c_1$

for $i = 0$ to $N-1$ **do**

$\quad r[i] \leftarrow (a[i] + b[i] + carry) \ \% \ 10$

$\quad carry \leftarrow (a[i] + b[i] + carry)/10$ $\quad c_2 N$

end for

$r[N] \leftarrow carry$ $\quad c_3$

The number of steps of this algorithm is $t(N) = (c_1 + c_3) + c_2 * N$.

When we analyze algorithms, we often ignore these constants.

Grade School Multiplication

for $j = 0$ to $N - 1$ **do**

  $carry \leftarrow 0$ N

  for $i = 0$ to $N - 1$ **do**

    $prod \leftarrow (a[i] * b[j] + carry)$

    $tmp[j][i + j] \leftarrow prod\%10$ N^2

    $carry \leftarrow prod/10$

  end for

  $tmp[j][N + j] \leftarrow carry$ N

end for

$carry \leftarrow 0$ 1

for $i = 0$ to $2 * N - 1$ **do**

  $sum \leftarrow carry$ N

  for $j = 0$ to $N - 1$ **do**

    $sum \leftarrow sum + tmp[j][i]$ N^2

  end for

  $r[i] \leftarrow sum\%10$ N

  $carry \leftarrow sum/10$

end for

Computational Complexity & O( )

We say…

Grade school addition takes time O(N), or
“big O of N” where N is the number of digits.

Grade school multiplication takes time O(N^2)
or “big O of N squared”.