Induction and Recursion 3

Today:

- ☐ Applications of the Fundamental Theorem of Arithmetic
- ☐ More recursion examples
- ☐ Solving recurrence relations

1 Applications of the Fundamental Theorem of Arithmetic

- How many (positive) divisors does $n = 2450250000 = 2^4 3^4 5^6 11^2$ have? Why, $5 \cdot 5 \cdot 7 \cdot 3 = 525$, as the divisors of n are the numbers of the for form $2^a 3^b 5^c 11^d$ where $a, b \in [0..4]$, $c \in [0..6]$, and $d \in [0..2]$.
- Is 1657728200 a square? No, because you can divide it by $10 \cdot 10 = 2^2 \cdot 5^2$ to get 16577282, which, if you divide it by two once more, will give you the odd number 8288641. So the number powers of 2 in our original number is 3. But a number is going to be a square iff all the powers of primes in its prime factorization are *even*. Can you see both directions of this?? Prove it!

2 Karatsuba Multiplication (1960/62)

Suppose we want to multiply two decimal numbers (binary numbers would work the same way). We write one number as $x = x_1 \| x_0$ and the other as $y = y_1 \| y_0$ with each half having m digits (let's not worry about what to do if m is odd; no significant complications are added). So

$$\begin{aligned} x &= x_1 10^m + x_0 \\ y &= y_1 10^m + y_0 \end{aligned}$$

The product is then

$$\begin{aligned} xy &= (x_1 \cdot 10^m + x_0)(y_1 \cdot 10^m + y_0) \\ &= z_2 \cdot 10^{2m} + z_1 \cdot 10^m + z_0 \end{aligned}$$

where

$$\begin{aligned} z_2 &= x_1 y_1 \\ z_1 &= x_1 y_0 + x_0 y_1 \\ z_0 &= x_0 y_0. \end{aligned}$$

Computing these values require *four* multiplications. Thus one way to multiply decomposes our size-n problem into four problems of size $n/2$, plus some added overhead that is $O(n)$:

$$T(n) = 4T(n/2) + n.$$

We will see shortly that decomposition does no better than grade-school multiplication.

Karatsuba observed that xy can be computed in only *three* multiplications of m-digit values: With z_0 and z_2 as before we can calculate z_1 by way of

$$z_1 = (x_1 + x_0)(y_1 + y_0) - z_2 - z_0 = (x_1 + x_0)(y_1 + y_0) - x_1 y_1 - x_0 y_0 = x_1 y_0 + x_0 y_1.$$

This give rise to the recurrence

$$T(n) = 3T(n/2) + n.$$

We will solve this recurrence in just a moment. For now, here is an example of how this works:

```
Example:
Let's compute
     98  76
  *  56  78
  ---------
       5928
     7644      These two numbers sum
     4256      to 11900, which we can also get as
   5488        = (98+76)(56+78) - 5928 - 5488
  ---------    =        174*134 - 5928 - 5488
   56075928    =          23316 - 5928 - 5488
               = 11900
```

3 Solving Recurrence Relations

Recurrence relations have the form $T(n) =$ an expressions involving values $T(k)$ where $k < n$; along with $T(k)$ =constant for sufficiently small k.

Let's figure out the running time $T(n)$ of Karatsuba multiply by the method of *repeated substitution*. Let $T(n)$ be the number of steps needed to multiply two n-bit (or n-digit) strings. We are only going to be seeking an answer that describes $T(n)$ within a constant, so we won't worry about exactly *what* we are counting, and we won't worry about ceilings and floors, either. Now

$$\begin{aligned}
T(n) &= 3T(n/2) + n \\
&= 3(3T(n/4) + n/2) + n \\
&= 3^2T(n/4) + 3n/2 + n \\
&= 3^2(3T(n/8 + n/4) + 3n/2 + n \\
&= 3^3T(n/8) + 3^2n/4 + 3n/2 + n \\
&= 3^4T(n/16) + n(1 + 3/2 + (3/2)^2 + (3/2)^3) \\
&= \cdots \\
&= 3^kT(n/2^k) + n(1 + 3/2 + (3/2)^2 + (3/2)^3 + \cdots + (2/3)^{k-1})
\end{aligned}$$

At this point it seems like it would be good to know what is

$$\begin{aligned}
S &= 1 + p + p^2 + \cdots + p^m \text{ and so} \\
Sp &= p + p^2 + \cdots + p^m + p^{m+1}. \text{ Subtracting,} \\
S - Sp &= 1 - p^{m+1} \text{ giving} \\
S(1-p) &= 1 - p^{m+1} \text{ and so} \\
S &= \frac{1 - p^{m+1}}{1-p} \text{ or} \\
S &= \frac{p^{m+1} - 1}{p - 1}.
\end{aligned}$$

In particular, with $p = 3/2$ we have

$$1 + 3/2 + \cdots + (3/2)^{k-1} = 2((3/2)^k - 1)$$

Going back to our goal of computing $T(n)$, we select $k = \lg n$ to conclude that

$$\begin{aligned}
T(n) &= 3^{\lg n} + 2n(3^{\lg n}/n - 1) \\
&= 3^{\lg n} + 2(3^{\lg n} - 2n \\
&= n^{\lg 3} + 2n^{\lg 3} - 2n \\
&\in \Theta(n^{\lg 3}) \\
&\subseteq O(n^{1.585})
\end{aligned}$$

Fastest known algorithm for this problem. It is possible to multiply two n-bit numbers in time $O(n \lg n)$. This is due to Harvey and van der Hoeven (2019). It follows a steady improvement in running times that begin with Karatsuba, continues with a famous result of Schönhage-Strassen (1971) that takes $O(n \log n \log \log n)$. The new $O(n \log n)$ multiplication algorithm is described at `https://tinyurl.com/3zhk24xe`.

In the remainder of these notes we'll look at other algorithms giving rise to such "divide-and-conquer" recurrence relations.

4 Binary Search

No doubt many of you have encountered this algorithm before. It is meant to determine if an element x is in an n-element list A of sorted elements—let's say in increasing (meaning non-decreasing) order. Like: $A = [-5, -2, 1, 3, 3, 7, 7, 12]$.

We first compare x with the element at the *middle* position p. When there is not middlemost position, go just to the left, say, of where the non-existent middle would be. If $x = A[p]$ you answer that, yes, x is in A. Otherwise, if $x < A[p]$ you should continue your search to the left of p. Otherwise, you should continue your search to the right of position p. You could write pseudocode like this:

algorithm BS(A, i, j, x)
if $i > j$ **then return** F
$p \leftarrow \lfloor (i + j)/2 \rfloor$
if $A[p] = x$ **then return** T
if $x < A[p]$ **then return** BS$(A, i, p - 1, x)$
if $x > A[p]$ **then return** BS$(A, p + 1, j, x)$

Now to analyze its running time:

$$T(n) = T(n/2) + 1$$

Use repeated substitution, as before, to get that $T(n) \in \Theta(\lg n)$.

5 Mergesort

This simple algorithm takes in a list A of n elements, indexed $A[1..n]$. The elements are drawn from a totally ordered universe—e.g., integers, reals, or strings. But they're in an arbitrary order. The algorithm returns a list with all of the same items but in increasing (meaning non-decreasing) order.

We employ a procedure Merge that takes in increasing-ordered lists L and R and returns a single list of $|L| + |R|$ whose elements are the elements appearing in L and R, but now in increasing order.

algorithm MS(A)
$n \leftarrow |A|$
if $n \leq 1$ **then return** A
$L \leftarrow \text{MS}(A[1..\lfloor n/2 \rfloor])$
$R \leftarrow \text{MS}(A[\lfloor n/2 \rfloor + 1..n])$
return Merge(L,R)

Analysis: Let $T(n)$ be the worst-case number of comparisons to sort n items using the algorithm above. Then $T(n) = 2T(n/2) + n - 1$. Show how to bound it by replacing the $n - 1$ with n and then using repeated substitution, as before, to get $T(n) \in \Theta(n \lg n)$.

Also show the recursion-tree view of solving the recurrence relation.