Algorithms: Divide and conquer arithmetic

Model 1: Arithmetic

$$\begin{array}{rcccccccc} & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ + & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}$$

$$\begin{array}{rcccccccccccc} & & & & & & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ & & & & & \times & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ \hline & & & & & & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ & & & & & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ & & & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ & & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ + & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{array}$$

In many situations, when we use arithmetic operations on integers such as addition or multiplication, we simply assume they take constant time. This is appropriate when the integers are bounded by some fixed size, *e.g.* if all the integers are 64-bit integers. However, if we want to be able to deal with integers of arbitrary size, this is no longer appropriate; we must take the size of the integers into account. For example, as of October 13, 2025, the largest known prime number is $2^{136\,279\,841} - 1$, a number with 41 024 320 decimal digits. The arithmetic operations needed to test this number for primality take a nontrivial amount of time!

Learning objective: Students will analyze arithmetic operations on n-bit integers.

1 In the addition operation shown in the model, how many bits long is each of the operands?

2 Approximately how many single bit operations (*e.g.* adding or comparing two bits) are needed to compute the addition $1011010_2 + 1110011_2$ shown in the model? Your answer should be a number,

but don't spend too much time arguing over the precise answer; make sure you agree to within ± 2 and move on.

3 In general, suppose we want to add two integers of n bits each. In terms of Θ, how many bit operations are needed?

4 Explain why it is not possible to do any better than this.

5 If we have a list of $\Theta(n)$ integers, each with $\Theta(n)$ bits, how long will it take (in terms of single bit operations) to add all of them?

Now consider the multiplication shown in the model.

6 Why are there five rows in between the two horizontal lines?

7 How many operations are needed to produce each such row? (*Hint*: you may assume that multiplying by a power of two takes constant time.)

8 If we are multiplying two integers with n bits each, how many intermediate rows could there be in the worst case?

9 How long will it take to add them all?

Model 2: Arithmetic by pieces

$X = 01101001_2$	$X_1 = 0110_2$	$X_2 = 1001_2$
$Y = 11100100_2$	$Y_1 = 1110_2$	$Y_2 = 0100_2$

Let's now consider whether it is possible to multiply two n-bit integers any faster.

10 What is the relationship between X, X_1 and X_2 in the model? What about Y, Y_1, and Y_2?

11 Suppose $Z = 1011100101_2$. What would Z_1 and Z_2 be?

12 What is $2^4 \cdot X_1$ in binary?

13 In general, if b is some number expressed in binary, what is $2^4 \cdot b$?

14 Write two equations expressing X in terms of X_1 and X_2 and Y in terms of Y_1 and Y_2.

15 In general, suppose A is an n-bit integer, and we split it into A_1 and A_2. Generalize your previous answer to express A in terms of A_1 and A_2.

16 Suppose A and B are n-bit integers, and consider the product AB. Expand both A and B using your previous answer, and distribute the resulting product. You should end up with an expression involving only A_1, A_2, B_1, and B_2.

17 How many multiplications are required to compute the expression from Question 16? (Remember that multiplying by a power of two takes constant time and does not need to be counted.)

18 How big (how many bits) are the inputs to each multiplication in Question 16?

19 Explain how we can use the equation from Question 16 as a recursive algorithm to compute AB.

20 Let $M(n)$ denote the time taken to multiply two n-bit integers, and write a recurrence relation for $M(n)$ corresponding to this recursive algorithm.