GUIDE TO REVIEW QUESTIONS FOR CHAPTER 3

1. **a)** See p. 168.
b) in English, in a computer language, in pseudocode
c) An algorithm is more of an abstract idea—a method in theory that will solve a problem. A computer program is the implementation of that idea into a specific syntactically correct set of instructions that a real computer can use to solve the problem. It is rather like the difference between a dollar (a certain amount of money, capable in theory of purchasing a certain quantity of goods and services) and a dollar bill.

2. **a)** See first three lines of text following Algorithm 1 on p. 169. **b)** See Algorithm 1 in Section 3.1.
c) See Example 1 in Section 3.3.

3. **a)** See p. 180. **b)** $n^2 + 18n + 107 \le n^3 + n^3 + n^3 = 3n^3$ for all $n > 107$.
c) n^3 is not less than a constant times $n^2 + 18n + 107$, since their ratio exceeds $n^3/(3n^2) = n/3$ for all $n > 107$.

4. **a)** For the sum, take the largest term; for the product multiply the factors together.
b) $g(n) = n^{n-2}2^n = (2n)^n/n^2$

5. **a)** the largest, average, and smallest number of comparisons used by the algorithm before it stops, among all lists of n integers
b) all are $n - 1$

6. **a)** See pp. 170–172. **b)** See pp. 194–195.
c) No—it depends on the lists involved. (However, the worst case complexity for binary search is always better than that for linear search for lists of any given size except for very short lists.)

7. **a)** See p. 173.
b) On the first pass, the 5 bubbles down to the end, producing 2 4 1 3 5. On the next pass, the 4 bubbles down to the end, producing 2 1 3 4 5. On the next pass, the 1 and the 2 are swapped. No further changes are made on the fourth pass.
c) $O(n^2)$; see Example 5 in Section 3.3

8. **a)** See p. 174.
b) On the first pass, the 5 is inserted into its correct position relative to the 2, producing 2 5 1 4 3. On the next pass, the 1 is inserted into its correct position relative to 2 5, producing 1 2 5 4 3. On the next pass, the 4 is inserted into its correct position relative to 1 2 5, producing 1 2 4 5 3. On the final pass, the 3 is inserted, producing the sorted list.
c) $O(n^2)$; see Example 6 in Section 3.3

9. **a)** See pp. 174–175. **b)** See Example 6 and Theorem 1 in Section 3.1.
c) See Exercise 55b in Section 3.1.

10. See p. 211.

11. **a)** See the bottom of p. 211. **b)** $11^2 \cdot 23 \cdot 29$

12. **a)** See p. 215.
b) find all the common factors (not a good algorithm unless the numbers are really small); find the prime factorization of each integer (works well if the numbers aren't too big and therefore can be easily factored); use the Euclidean algorithm (really the best method)
c) 1 (use the Euclidean algorithm)
d) $2^3 3^5 5^5 7^3$

13. **a)** $7 \mid a-b$ **b)** $0 \equiv -7$; $-1 \equiv -8$; $3 \equiv 17 \equiv -11$
c) $(10a+13)-(-4b+20)=3(a-b)+7(a+b-1)$; note that 7 divides both terms

14. See Algorithm 1 in Section 3.6 with $b=16$. Use "A" for a remainder of 10, "B" for a remainder of 11, and so on. See also Example 4 in that section.

15. **a)** Use the Euclidean algorithm; see Example 1 in Section 3.7.
b) $7 = 5 \cdot 119 - 7 \cdot 84$

16. **a)** $a\overline{a} \equiv 1 \pmod{m}$
b) Express 1 as $sa+tm$ (see Review Question 15). Then s is the inverse of a modulo m.
c) 11

17. **a)** Multiply each side by the inverse of a modulo m. **b)** $\{\,10+19k \mid k \in \mathbf{Z}\,\}$

18. **a)** See p. 236. **b)** $\{\,17+140k \mid k \in \mathbf{Z}\,\}$

19. No—n could be a pseudoprime such as 341.

20. **a)** See p. 241.
b) The amount of shift, k, is kept secret. It is needed both to encode and to decode messages.
c) Although the key for decoding, d, is kept secret, the keys for encoding, n and e, are published.

21. See p. 248. The number of columns of $\mathbf{A}$ must equal the number of rows of $\mathbf{B}$.

22. **a)** 5 **b)** $(\mathbf{A}_1\mathbf{A}_2)(\mathbf{A}_3\mathbf{A}_4)$; see Exercise 25 in Section 3.8

SUPPLEMENTARY EXERCISES FOR CHAPTER 3

1. **a)** This algorithm will be identical to the algorithm *first largest* for Exercise 17 of Section 3.1, except that we want to change the value of *location* each time we find another element in the list that is equal to the current value of *max*. Therefore we simply change the strict less than ($<$) in the comparison $max < a_i$ to a less than or equal to ($\leq$), rendering the fifth line of that procedure "**if** $max \leq a_i$ **then**."
b) The number of comparisons used by this algorithm can be computed as follows. There are $n-1$ passes through the **for** loop, each one requiring a comparison of max with a_i. In addition, n comparisons are needed for bookkeeping for the loop (comparison of i with n, as i assumes the values $2, 3, \ldots, n+1$). Therefore $2n-1$ comparisons are needed altogether, which is $O(n)$.

3. **a)** We will try to write an algorithm sophisticated enough to avoid unnecessary checking. The answer—**true** or **false**—will be placed in a variable called *zeros*, the name of the procedure. (This approach is used by function subprograms in some computer languages, although it is not always permitted to use the procedure name in an expression as we do here.)

```
**procedure** $zeros(a_1a_2\ldots a_n$ : bit string)
$i := 1$
$zeros$ := **false** {no pair of zeros found yet}
**while** $i < n$ and $\neg zeros$
        **if** $a_i = 1$ **then** $i := i + 1$
        **else if** $a_{i+1} = 1$ **then** $i := i + 2$
        **else** $zeros$ := **true**
{ $zeros$ was set to **true** if and only if there were a pair of consecutive zeros}
```

b) The number of comparisons depends on whether a pair of 0's is found and also depends on the pattern of increments of the looping variable i. Without getting into the intricate details of exactly which is the worst case, we note that at worst there are approximately n passes through the loop, each requiring one comparison of a_i with 1 (there may be two comparisons on some passes, but then there will be fewer passes). In addition, n bookkeeping comparisons of i with n are needed (we are ignoring the testing of the logical variable $zeros$). Thus a total of approximately $2n$ comparisons are used, which is $O(n)$.

5. **a)** and **b)**. We have a variable min to keep track of the minimum as well as a variable max to keep track of the maximum.

```
**procedure** $smallest\ and\ largest(a_1, a_2, \ldots, a_n$ : integers)
$min := a_1$
$max := a_1$
**for** $i := 2$ **to** $n$
**begin**
        **if** $a_i < min$ **then** $min := a_i$
        **if** $a_i > max$ **then** $max := a_i$
**end** { $min$ is the smallest integer among the input, and $max$ is the largest}
```

c) There are two comparisons for each iteration of the loop, and there are $n-1$ iterations, so there are $2n-2$ comparisons in all.

7. We think of ourselves as observers as some algorithm for solving this problem is executed. We do not care what the algorithm's strategy is, but we view it along the following lines, in effect taking notes as to what is happening and what *we* know as it proceeds. Before any comparisons are done, there is a possibility that each element could be the maximum and a possibility that it could be the minimum. This means that there are $2n$ different possibilities, and $2n-2$ of them have to be eliminated through comparisons of elements, since we need to find the unique maximum and the unique minimum. We classify comparisons of two elements as "nonvirgin" or "virgin," depending on whether or not both elements being compared have been in any previous comparison. A virgin comparison, between two elements that have not yet been involved in any comparison, eliminates the possibility that the larger one is the minimum and that the smaller one is the maximum; thus each virgin comparison eliminates two possibilities, but it clearly cannot do more. A nonvirgin comparison, one involving at least one element that has been compared before, must be between two elements that are still in the running to be the maximum or two elements that are still in the running to be the minimum, and at least one of these elements must *not* be in the running for the other category. For example, we might be comparing x and y, where all we know is that x has been eliminated as the minimum. If we find that $x > y$ in this case, then only one possibility has been ruled out—we now know that y is not the maximum. Thus in the worst case, a nonvirgin comparison eliminates only one possibility. (The cases of other nonvirgin comparisons are similar.) Now there are at most $\lfloor n/2 \rfloor$ comparisons of elements that have never been compared before, each removing two possibilities; they remove $2\lfloor n/2 \rfloor$ possibilities altogether. Therefore we need $2n-2-2\lfloor n/2 \rfloor$ more comparisons that, as we have argued, can remove only one possibility each, in order to find the answers in the worst case, since $2n-2$ possibilities have to be eliminated. This gives us a total of $2n-2-2\lfloor n/2 \rfloor + \lfloor n/2 \rfloor$ comparisons in all. But $2n-2-2\lfloor n/2 \rfloor+\lfloor n/2 \rfloor = 2n-2-\lfloor n/2 \rfloor = 2n-2+\lceil -n/2 \rceil = \lceil 2n-n/2 \rceil -2 = \lceil 3n/2 \rceil -2$, as desired.

Note that this gives us a lower bound on the number of comparisons used in an algorithm to find the minimum and the maximum. On the other hand, Exercise 6 gave us an upper bound of the same size. Thus the algorithm in Exercise 6 is the most efficient algorithm possible for solving this problem.

9. After the comparison and possible exchange of adjacent elements on the first pass, from front to back, the list is $3, 1, 4, 5, 2, 6$, where the 6 is known to be in its correct position. After the comparison and possible exchange of adjacent elements on the second pass, from back to front, the list is $1, 3, 2, 4, 5, 6$, where the 6 and the 1 are known to be in their correct positions. After the next pass, the result is $1, 2, 3, 4, 5, 6$. One more pass finds nothing to exchange, and the algorithm terminates.

11. There are possibly as many as $n-1$ passes through the list (or parts of it—it depends on the particular way it is implemented), and each pass uses $O(n)$ comparisons. Thus there are $O(n^2)$ comparisons in all.

13. Since $\log n < n$, we have $(n \log n + n^2)^3 \le (n^2 + n^2)^3 \le (2n^2)^3 = 8n^6$ for all $n > 0$. This proves that $(n \log n + n^2)^3$ is $O(n^6)$, with witnesses $C = 8$ and $k = 0$.

15. In the first factor the x^2 term dominates the other term, since $(\log x)^3$ is $O(x)$. Therefore by Theorem 2 in Section 3.2, this term is $O(x^2)$. Similarly, in the second factor, the 2^x term dominates. Thus by Theorem 3 of Section 3.2, the product is $O(x^2 2^x)$.

17. Let us look at the ratio

$$\frac{n!}{2^n} = \frac{n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1}{2 \cdot 2 \cdot 2 \cdots 2 \cdot 2 \cdot 2} = \frac{n}{2} \cdot \frac{n-1}{2} \cdot \frac{n-2}{2} \cdots \frac{3}{2} \cdot \frac{2}{2} \cdot \frac{1}{2}.$$

Each of the fractions in the final expression is greater than 1 except the last one, so the entire expression is at least $(n/2)/2 = n/4$. Since $n!/2^n$ increases without bound as n increases, $n!$ cannot be bounded by a constant times 2^n, which tells us that $n!$ is not $O(2^n)$.

19. Obviously there are an infinite number of possible answers. The numbers congruent to 5 modulo 17 include $5, 22, 39, 56, \ldots$, as well as $-12, -29, -46, \ldots$.

21. From the hypothesis $ac \equiv bc \pmod{m}$ we know that $ac - bc = km$ for some integer k. Divide both sides by c to obtain the equation $a - b = (km)/c$. Now the left-hand side is an integer, and so the right-hand side must be an integer as well. In other words, $c \mid km$. Letting $d = \gcd(m, c)$, we write $c = de$. Then the way that c divides km is that $d \mid m$ and $e \mid k$ (since no factor of e divides m/d). Thus our equation reduces to $a - b = (k/e)(m/d)$, where both factors on the right are integers. By definition, this means that $a \equiv b \pmod{m/d}$.

23. $\gcd(10223, 33341) = \gcd(10223, 2672) = \gcd(2672, 2207) = \gcd(2207, 465) = \gcd(465, 347) = \gcd(347, 118) = \gcd(118, 111) = \gcd(111, 7) = \gcd(7, 6) = \gcd(6, 1) = \gcd(1, 0) = 1$

25. By Lemma 1 in Section 3.6, $\gcd(2n+1, 3n+2) = \gcd(2n+1, n+1)$, since $2n+1$ goes once into $3n+2$ with a remainder of $n+1$. Now if we divide $n+1$ into $2n+1$, we get a remainder of n, so the answer must equal $\gcd(n+1, n)$. At this point, the remainder when dividing n into $n+1$ is 1, so the answer must equal $\gcd(n, 1)$, which is clearly 1. Thus the answer is 1.

27. This problem is similar to Exercise 7 in Section 3.5. Without loss of generality, we may assume that the given integer is positive (since $n \mid a$ if and only if $n \mid (-a)$, and the case $a = 0$ is trivial). Let the decimal expansion of the integer a be given by $a = (a_{n-1}a_{n-2} \ldots a_1 a_0)_{10}$. Thus $a = 10^{n-1}a_{n-1} + 10^{n-2}a_{n-2} + \cdots + 10a_1 + a_0$. Since $10 \equiv 1 \pmod{9}$, we have $a \equiv a_{n-1} + a_{n-2} + \cdots + a_1 + a_0 \pmod{9}$. Therefore $a \equiv 0 \pmod{9}$ if and only if the sum of the digits is congruent to $0 \pmod 9$. Since being divisible by 9 is the same as being congruent to $0 \pmod 9$, we have proved that an integer is divisible by 9 if and only if the sum of its decimal digits is divisible by 9.

29. It might be helpful to read the solution to Exercise 35 in Section 3.5 to see the philosophy behind this approach. Suppose by way of contradiction that $q_1, q_2, \ldots, q_n$ are all the primes of the form $6k+5$. Thus $q_1 = 5$, $q_2 = 11$, and so on. Let $Q = 6q_1q_2\cdots q_n - 1$. We note that Q is of the form $6k+5$, where $k = q_1q_2\cdots q_n - 1$. Now Q has a prime factorization $Q = p_1p_2\cdots p_t$. Clearly no p_i is 2, 3, or any q_j, because the remainder when Q is divided by 2 is 1, by 3 is 2, and by q_j is $q_j - 1$. All odd primes other than 3 are of the form $6k+1$ or $6k+5$, and the product of primes of the form $6k+1$ is also of this form. Therefore at least one of the p_i's must be of the form $6k+5$, a contradiction.

31. **a)** Since 2 is a factor of all three of these integers, this set is not mutually relatively prime.
b) Since 12 and 25 share no common factors, this set has greatest common divisor 1, so it is mutually relatively prime. (It is possible for every pair of integers in a set of mutually relatively prime integers to have a nontrivial common factor (see Exercise 32), but certainly if two of the integers in a set are relatively prime, then the set is automatically mutually relatively prime.)
c) Since 15 and 28 share no common factors, this set has greatest common divisor 1, so it is mutually relatively prime.
d) Since 21 and 32 share no common factors, this set has greatest common divisor 1, so it is mutually relatively prime.

33. **a)** We need to find the inverse function. In other words, given $(ap+b)$ **mod** 26, how does one recover p? Working modulo 26, if we subtract b, then we will have ap. If we then multiply by an inverse of a (which must exist since we are assuming that $\gcd(a, 26) = 1$), we will have p back. Therefore the decryption function is $g(q) = \overline{a}(q-b)$ **mod** 26, where $\overline{a}$ is an inverse of a modulo 26.
b) Translating the letters into numbers, we see that the message is 11–9–12–10–6 12–6–12–23–5 16–4–23–12–22. By part **(a)** the decryption function is $g(q) = 15(q-10)$ **mod** 26. (This required computing the inverse of 7 modulo 26, using the technique shown in the solution to Exercise 7 in Section 3.7; it is easy to see that $7 \cdot 15 = 105 \equiv 1 \pmod{26}$.) Applying this function, we obtain the decrypted message 15–11–4–0–18 4–18–4–13–3 12–14–13–4–24. This translates into PLEAS ESEND MONEY, which, after correcting the spacing, is "please send money."

35. The least common multiple of 6 and 15 is 30, so the set of solutions will be given modulo 30 (see Exercise 36). Since the numbers involved here are so small, it is probably best simply to write down the solutions of $x \equiv 4 \pmod 6$ and then see which, if any, of them are also solutions of $x \equiv 13 \pmod{15}$. The solutions of the first congruence, up to 30, are 4, 10, 16, 22, and 28. Only 28 is congruent to 13 modulo 15. Therefore the general solution is all numbers congruent to 28 modulo 30, i.e., $\ldots, -32, -2, 28, 58, \ldots$.

37. We give a proof by contradiction. For this proof we need a fact about polynomials, namely that a nonconstant polynomial can take on the same value only a finite number of times. (Think about its graph—a polynomial of degree n has at most n "wiggles" and so a horizontal line can intersect it at most n times. Alternatively, this statement follows from the Fundamental Theorem of Algebra, which guarantees that a polynomial of degree n has at most n 0's; if $f(x) = a$, then x is a zero of the polynomial $f(x) - a$.) Our given polynomial f can take on the values 0 and ± 1 only finitely many times, so if there is not some y such that $f(y)$ is composite, then there must be some x_0 such that $\pm f(x_0)$ is prime, say p. Now look at $f(x_0 + kp)$. When we plug $x_0 + kp$ in for x in the polynomial and multiply it out, every term will contain a factor of p except for the terms that form $f(x_0)$. Therefore $f(x_0 + kp) = f(x_0) + mp = (m \pm 1)p$ for some integer m. As k varies, this value can be 0 or p or $-p$ only finitely many times; therefore it must be a different multiple of p and therefore a composite number for some values of k, and our proof is complete.

39. Assume that every even integer greater than 2 is the sum of two primes, and let n be an integer greater than 5. If n is odd, then we can write $n = 3 + (n-3)$, decompose $n - 3 = p + q$ into the sum of two primes (since $n - 3$ is an even integer greater than 2), and therefore have written $n = 3 + p + q$ as the sum of three primes. If n is even, then we can write $n = 2 + (n-2)$, decompose $n - 2 = p + q$ into the sum of two primes (since $n - 2$ is an even integer greater than 2), and therefore have written $n = 2 + p + q$ as the sum of three primes. For the converse, assume that every integer greater than 5 is the sum of three primes, and let n be an even integer greater than 2. By our assumption we can write $n + 2$ as the sum of three primes. Since $n + 2$ is even, these three primes cannot all be odd, so we have $n + 2 = 2 + p + q$, where p and q are primes, whence $n = p + q$, as desired.

41. Let us begin by computing $\mathbf{A}^n$ for the first few values of n.

$$\mathbf{A}^1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \mathbf{A}^2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{A}^3 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{A}^4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Since $\mathbf{A}^4 = \mathbf{I}$, the pattern will repeat from here: $\mathbf{A}^5 = \mathbf{A}^4\mathbf{A} = \mathbf{IA} = \mathbf{A}$, $\mathbf{A}^6 = \mathbf{A}^2$, $\mathbf{A}^7 = \mathbf{A}^3$, and so on. Thus for all $n \geq 0$ we have

$$\mathbf{A}^{4n+1} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \mathbf{A}^{4n+2} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{A}^{4n+3} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{A}^{4n+4} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

43. (The notation $c\mathbf{I}$ means the identity matrix $\mathbf{I}$ with each entry multiplied by the real number c; thus this matrix consists of c's along the main diagonal and 0's elsewhere.) Let $\mathbf{A} = \begin{bmatrix} u & v \\ w & x \end{bmatrix}$. We will determine what these entries have to be by using the fact that $\mathbf{AB} = \mathbf{BA}$ for a few judiciously chosen matrices $\mathbf{B}$. First let $\mathbf{B} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$. Then $\mathbf{AB} = \begin{bmatrix} 0 & u \\ 0 & w \end{bmatrix}$, and $\mathbf{BA} = \begin{bmatrix} w & x \\ 0 & 0 \end{bmatrix}$. Since these two must be equal, we know that $0 = w$ and $u = x$. Next choose $\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$. Then we get $\begin{bmatrix} v & 0 \\ x & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ u & v \end{bmatrix}$, whence $v = 0$. Therefore the matrix $\mathbf{A}$ must be in the form $\begin{bmatrix} u & 0 \\ 0 & u \end{bmatrix}$, which is just u times the identity matrix, as desired.

45. In an $n \times n$ upper-triangular matrix, all entries a_{ij} are zero unless $i \leq j$. Therefore we can store such matrices in about half the space that would be required to store an ordinary $n \times n$ matrix. In implementing something like Algorithm 1 in Section 3.8, then, we need only do the computations for those values of the indices that can produce nonzero entries. The following algorithm does this. We follow the usual notation: $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$.

```
procedure triangular matrix multiplication(A, B : upper-triangular matrices)
for $i := 1$ to $n$
    for $j := i$ to $n$ {since we want $j \geq i$}
    begin
        $c_{ij} := 0$
        for $k := i$ to $j$ {the only relevant part}
            $c_{ij} := c_{ij} + a_{ik}b_{kj}$
    end
{the upper-triangular matrix $\mathbf{C} = [c_{ij}]$ is the product of $\mathbf{A}$ and $\mathbf{B}$}
```

47. We simply need to show that the alleged inverse of $\mathbf{AB}$ has the correct defining property—that its product with $\mathbf{AB}$ (on either side) is the identity. Thus we compute

$$(\mathbf{AB})(\mathbf{B}^{-1}\mathbf{A}^{-1}) = \mathbf{A}(\mathbf{BB}^{-1})\mathbf{A}^{-1} = \mathbf{AIA}^{-1} = \mathbf{AA}^{-1} = \mathbf{I},$$

and similarly $(\mathbf{B}^{-1}\mathbf{A}^{-1})(\mathbf{AB}) = \mathbf{I}$. Therefore $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$. (Note that the indicated matrix multiplications were all defined, since the hypotheses implied that both $\mathbf{A}$ and $\mathbf{B}$ were $n \times n$ matrices for some (and the same) n.)

49. **a)** The $(i,j)^{\text{th}}$ entry of $\mathbf{A} \odot \mathbf{0}$ is by definition the Boolean sum ($\vee$) of some Boolean products ($\wedge$) of the form $a_{ik} \wedge 0$. Since the latter always equals 0, every entry is 0, so $\mathbf{A} \odot \mathbf{0} = \mathbf{0}$. Similarly $\mathbf{0} \odot \mathbf{A}$ consists of entries that are all 0, so it, too, equals $\mathbf{0}$.
b) Since $\vee$ operates entrywise, the statements that $\mathbf{A} \vee \mathbf{0} = \mathbf{A}$ and $\mathbf{0} \vee \mathbf{A} = \mathbf{A}$ follow from the facts that $a_{ij} \vee 0 = a_{ij}$ and $0 \vee a_{ij} = a_{ij}$.
c) Since $\wedge$ operates entrywise, the statements that $\mathbf{A} \wedge \mathbf{0} = \mathbf{0}$ and $\mathbf{0} \wedge \mathbf{A} = \mathbf{0}$ follow from the facts that $a_{ij} \wedge 0 = 0$ and $0 \wedge a_{ij} = 0$.

51. We assume that someone has chosen a positive integer less than 2^n, which we are to guess. We ask the person to write the number in binary, using leading 0's if necessary to make it n bits long. We then ask "Is the first bit a 1?", "Is the second bit a 1?", "Is the third bit a 1?", and so on. After we know the answers to these n questions, we will know the number, because we will know its binary expansion.

WRITING PROJECTS FOR CHAPTER 3

Books and articles indicated by bracketed symbols below are listed near the end of this manual. You should also read the general comments and advice you will find there about researching and writing these essays.

1. Algorithms are as much a subject of computer science as they are a subject of mathematics. Thus sources on the history of computer science (or perhaps even verbose introductory programming or comprehensive computer science textbooks) might be a good place to look. See also [Ha2].

2. The original is [Ba2]. Good history of mathematics books would be a place to follow up.

3. See [Me1], or do a Web search.

4. Knuth's volume 3 [Kn] is the bible for sorting algorithms.

5. This is a vast and extremely important subject in computer science today. One basic book on the subject is [Gi]; see also an essay in [De2].

6. See the comments for Writing Project 5, above. One basic algorithm to think about doing in parallel is sorting. Imagine a group of school-children asked to arrange themselves in a row by height, and any child can determine whether another child is shorter or taller than him/herself.

7. As usual, the Web is an excellent resource here. Check out the GIMPS page and then follow its links: `http://www.mersenne.org/prime.htm`.

8. These primes are sometimes jokingly referred to as "industrial strength primes." Number theory textbooks, such as [Ro3], would be a place to start. See also the article [Le3].

9. One can often find mathematical news reported in *The New York Times* and other nontechnical media. Search an index to find a story about this topic. (While you're looking at back issues of the *Times*, read the January 31, 1995, article on the solution to Fermat's Last Theorem.)

10. Your essay should mention the RSA-129 project. See *The New York Times*, around the spring of 1994 (use its index). For an expository article, try [Po].

11. There are dozens of books on computer hardware and circuit design that discuss the algorithms and circuits used in performing these operations. If you are a computer science or computer engineering major, you probably have taken (or will take) a course that deals with these topics. See [Ko2] and similar books.

12. A traditional history of mathematics book should be helpful here; try [Bo4] or [Ev3].

13. This topic bas taken on a lot of significance recently as randomized algorithms become more and more important. The August/September 1994 issue of *SIAM News* (the newsletter of the Society for Industrial and Applied Mathematics) has a provocative article on the subject. See also [La1], or for older material in a textbook try volume 2 of [Kn].

14. This topic gets into the news on a regular basis. Try *The New York Times* index. See also Writing Project 10, above, and 15, below.

15. If I encrypt my signature with my private key, then I will produce something that will decrypt (using my public key) as my signature. Furthermore, no one else can do this, since no one else knows my private key. Good sources for cryptography include [Be], [MeOo], and [St2].

16. The author's number theory text ([Ro3]) has material on this topic. The amazing mathematician John H. Conway (inventor of the Game of Life, among other things) has devised what he calls the Doomsday Algorithm, and it works quite fast with practice. See [BeCo]. (Conway can determine any day of the week mentally in a second or two.)

17. An advanced algorithms text, such as [Ma2] or [BrBr], is the place to look. One algorithm has the names Schönhage and Strassen associated with it. A related topic is the fast Fourier transform. See also volume 2 of [Kn].

18. This is related to Writing Project 17; see the suggestions there. Strassen also invented (or discovered, depending on your philosophy) a fast matrix multiplication algorithm.

19. Several excellent books have appeared in the past decade on cryptography, such as [MeOo]. Many of them, including that one, will treat this topic.

CHAPTER 4
Induction and Recursion

SECTION 4.1 Mathematical Induction

Understanding and constructing proofs by mathematical induction are extremely difficult tasks for most students. Do not be discouraged, and do not give up, because, without doubt, this proof technique is the most important one there is in mathematics and computer science. Pay careful attention to the conventions to be observed in writing down a proof by induction. As with all proofs, remember that a proof by mathematical induction is like an essay it must have a beginning, a middle, and an end; it must consist of complete sentences, logically and aesthetically arranged; and it must convince the reader. Be sure that your basis step (also called the "base case") is correct (that you have verified the proposition in question for the smallest value or values of n), and be sure that your inductive step is correct and complete (that you have derived the proposition for $k+1$, assuming the inductive hypothesis that the proposition is true for k).

Some, but not all, proofs by mathematical induction are like Exercises 3 17. In each of these, you are asked to prove that a certain summation has a "closed form" representation given by a certain expression. Here the proofs are usually straightforward algebra. For the inductive step you start with the summation for $P(k+1)$, find the summation for $P(k)$ as its first k terms, replace that much by the closed form given by the inductive hypothesis, and do the algebra to get the resulting expression into the desired form. When doing proofs like this, however, remember to include all the words surrounding your algebra– the algebra alone is not the proof. Also keep in mind that $P(n)$ is the proposition that the sum equals the closed-form expression, not just the sum and not just the expression.

Many inequalities can be proved by mathematical induction; see Exercises 18 24, for example. The method also extends to such things as set operations, divisibility, and a host of other applications; a sampling of them is given in other exercises in this set. Some are quite complicated.

One final point about notation. In performing the inductive step, it really does not matter what letter we use. We see in the text the proof of $P(k) \to P(k+1)$; but it would be just as valid to prove $P(n) \to P(n+1)$, since the k in the first case and the n in the second case are just dummy variables. We will use both notations in this Guide; in particular, we will use k for the first few exercises but often use n afterwards.

1. We can prove this by mathematical induction. Let $P(n)$ be the statement that the train stops at station n. We want to prove that $P(n)$ is true for all positive integers n. For the basis step, we are told that $P(1)$ is true. For the inductive step, we are told that $P(k)$ implies $P(k+1)$ for each $k \geq 1$. Therefore by the principle of mathematical induction, $P(n)$ is true for all positive integers n.

3. **a)** Plugging in $n=1$ we have that $P(1)$ is the statement $1^2 = 1 \cdot 2 \cdot 3/6$.

b) Both sides of $P(1)$ shown in part **(a)** equal 1.

c) The inductive hypothesis is the statement that

$$1^2 + 2^2 + \cdots + k^2 = \frac{k(k+1)(2k+1)}{6}.$$

d) For the inductive step, we want to show for each $k \geq 1$ that $P(k)$ implies $P(k+1)$. In other words, we