SECTION 4.3 Recursive Definitions and Structural Induction

2. a) $f(1) = -2f(0) = -2 \cdot 3 = -6$, $f(2) = -2f(1) = -2 \cdot (-6) = 12$, $f(3) = -2f(2) = -2 \cdot 12 = -24$, $f(4) = -2f(3) = -2 \cdot (-24) = 48$, $f(5) = -2f(4) = -2 \cdot 48 = -96$

b) $f(1) = 3f(0) + 7 = 3 \cdot 3 + 7 = 16$, $f(2) = 3f(1) + 7 = 3 \cdot 16 + 7 = 55$, $f(3) = 3f(2) + 7 = 3 \cdot 55 + 7 = 172$, $f(4) = 3f(3) + 7 = 3 \cdot 172 + 7 = 523$, $f(5) = 3f(4) + 7 = 3 \cdot 523 + 7 = 1576$

c) $f(1) = f(0)^2 - 2f(0) - 2 = 3^2 - 2 \cdot 3 - 2 = 1$, $f(2) = f(1)^2 - 2f(1) - 2 = 1^2 - 2 \cdot 1 - 2 = -3$, $f(3) = f(2)^2 - 2f(2) - 2 = (-3)^2 - 2 \cdot (-3) - 2 = 13$, $f(4) = f(3)^2 - 2f(3) - 2 = 13^2 - 2 \cdot 13 - 2 = 141$, $f(5) = f(4)^2 - 2f(4) - 2 = 141^2 - 2 \cdot 141 - 2 = 19{,}597$

d) First note that $f(1) = 3^{f(0)/3} = 3^{3/3} = 3 = f(0)$. In the same manner, $f(n) = 3$ for all n.

4. a) $f(2) = f(1) - f(0) = 1 - 1 = 0$, $f(3) = f(2) - f(1) = 0 - 1 = -1$, $f(4) = f(3) - f(2) = -1 - 0 = -1$, $f(5) = f(4) - f(3) = -1 - 1 = 0$

b) Clearly $f(n) = 1$ for all n, since $1 \cdot 1 = 1$.

c) $f(2) = f(1)^2 + f(0)^3 = 1^2 + 1^3 = 2$, $f(3) = f(2)^2 + f(1)^3 = 2^2 + 1^3 = 5$, $f(4) = f(3)^2 + f(2)^3 = 5^2 + 2^3 = 33$, $f(5) = f(4)^2 + f(3)^3 = 33^2 + 5^3 = 1214$

d) Clearly $f(n) = 1$ for all n, since $1/1 = 1$.

6. a) This is valid, since we are provided with the value at $n = 0$, and each subsequent value is determined by the previous one. Since all that changes from one value to the next is the sign, we conjecture that $f(n) = (-1)^n$. This is true for $n = 0$, since $(-1)^0 = 1$. If it is true for $n = k$, then we have $f(k+1) = -f(k+1-1) = -f(k) = -(-1)^k$ by the inductive hypothesis, whence $f(k+1) = (-1)^{k+1}$.

b) This is valid, since we are provided with the values at $n = 0$, 1, and 2, and each subsequent value is determined by the value that occurred three steps previously. We compute the first several terms of the sequence: $1, 0, 2, 2, 0, 4, 4, 0, 8, \ldots$. We conjecture the formula $f(n) = 2^{n/3}$ when $n \equiv 0 \pmod 3$, $f(n) = 0$ when $n \equiv 1 \pmod 3$, $f(n) = 2^{(n+1)/3}$ when $n \equiv 2 \pmod 3$. To prove this, first note that in the base cases we have $f(0) = 1 = 2^{0/3}$, $f(1) = 0$, and $f(2) = 2 = 2^{(2+1)/3}$. Assume the inductive hypothesis that the formula is valid for smaller inputs. Then for $n \equiv 0 \pmod 3$ we have $f(n) = 2f(n-3) = 2 \cdot 2^{(n-3)/3} = 2 \cdot 2^{n/3} \cdot 2^{-1} = 2^{n/3}$, as desired. For $n \equiv 1 \pmod 3$ we have $f(n) = 2f(n-3) = 2 \cdot 0 = 0$, as desired. And for $n \equiv 2 \pmod 3$ we have $f(n) = 2f(n-3) = 2 \cdot 2^{(n-3+1)/3} = 2 \cdot 2^{(n+1)/3} \cdot 2^{-1} = 2^{(n+1)/3}$, as desired.

c) This is invalid. We are told that $f(2)$ is defined in terms of $f(3)$, but $f(3)$ has not been defined.

d) This is invalid, because the value at $n = 1$ is defined in two conflicting ways—first as $f(1) = 1$ and then as $f(1) = 2f(1-1) = 2f(0) = 2 \cdot 0 = 0$.

e) This appears syntactically to be not valid, since we have conflicting instruction for odd $n \geq 3$. On the one hand $f(3) = f(2)$, but on the other hand $f(3) = 2f(1)$. However, we notice that $f(1) = f(0) = 2$ and $f(2) = 2f(0) = 4$, so these apparently conflicting rules tell us that $f(3) = 4$ on the one hand and $f(3) = 2 \cdot 2 = 4$ on the other hand. Thus we got the same answer either way. Let us show that in fact this definition is valid because the rules coincide.

We compute the first several terms of the sequence: $2, 2, 4, 4, 8, 8, \ldots$. We conjecture the formula $f(n) = 2^{\lceil (n+1)/2 \rceil}$. To prove this inductively, note first that $f(0) = 2 = 2^{\lceil (0+1)/2 \rceil}$. For larger values we have for n odd using the first part of the recursive step that $f(n) = f(n-1) = 2^{\lceil (n-1+1)/2 \rceil} = 2^{\lceil n/2 \rceil} = 2^{\lceil (n+1)/2 \rceil}$, since $n/2$ is not an integer. For $n \geq 2$, whether even or odd, using the second part of the recursive step we have $f(n) = 2f(n-2) = 2 \cdot 2^{\lceil (n-2+1)/2 \rceil} = 2 \cdot 2^{\lceil (n+1)/2 \rceil - 1} = 2 \cdot 2^{\lceil (n+1)/2 \rceil} \cdot 2^{-1} = 2^{\lceil (n+1)/2 \rceil}$, as desired.

8. Many answers are possible.

a) Each term is 4 more than the term before it. We can therefore define the sequence by $a_1 = 2$ and $a_{n+1} = a_n + 4$ for all $n \geq 1$.

b) We note that the terms alternate: 0, 2, 0, 2, and so on. Thus we could define the sequence by $a_1 = 0$, $a_2 = 2$, and $a_n = a_{n-2}$ for all $n \geq 3$.

c) The sequence starts out 2, 6, 12, 20, 30, and so on. The differences between successive terms are 4, 6, 8, 10, and so on. Thus the n^{th} term is $2n$ greater than the term preceding it; in symbols: $a_n = a_{n-1} + 2n$. Together with the initial condition $a_1 = 2$, this defines the sequence recursively.

d) The sequence starts out 1, 4, 9, 16, 25, and so on. The differences between successive terms are 3, 5, 7, 9, and so on—the odd numbers. Thus the n^{th} term is $2n-1$ greater than the term preceding it; in symbols: $a_n = a_{n-1} + 2n - 1$. Together with the initial condition $a_1 = 1$, this defines the sequence recursively.

10. The base case is that $S_m(0) = m$. The recursive part is that $S_m(n+1)$ is the successor of $S_m(n)$ (i.e., the integer that follows $S_m(n)$, namely $S_m(n)+1$).

12. The basis step ($n = 1$) is clear, since $f_1^2 = f_1 f_2 = 1$. Assume the inductive hypothesis. Then $f_1^2 + f_2^2 + \cdots + f_n^2 + f_{n+1}^2 = f_n f_{n+1} + f_{n+1}^2 = f_{n+1}(f_n + f_{n+1}) = f_{n+1} f_{n+2}$, as desired.

14. The basis step ($n = 1$) is clear, since $f_2 f_0 - f_1^2 = 1 \cdot 0 - 1^2 = -1 = (-1)^1$. Assume the inductive hypothesis. Then we have

$$\begin{aligned} f_{n+2} f_n - f_{n+1}^2 &= (f_{n+1} + f_n) f_n - f_{n+1}^2 \\ &= f_{n+1} f_n + f_n^2 - f_{n+1}^2 \\ &= -f_{n+1}(f_{n+1} - f_n) + f_n^2 \\ &= -f_{n+1} f_{n-1} + f_n^2 \\ &= -(f_{n+1} f_{n-1} - f_n^2) \\ &= -(-1)^n = (-1)^{n+1}. \end{aligned}$$

16. The basis step ($n = 1$) is clear, since $f_0 - f_1 + f_2 = 0 - 1 + 1 = 0$, and $f_1 - 1 = 0$ as well. Assume the inductive hypothesis. Then we have (substituting using the defining relation for the Fibonacci sequence where appropriate)

$$\begin{aligned} f_0 - f_1 + f_2 - \cdots - f_{2n-1} + f_{2n} - f_{2n+1} + f_{2n+2} &= f_{2n-1} - 1 - f_{2n+1} + f_{2n+2} \\ &= f_{2n-1} - 1 + f_{2n} \\ &= f_{2n+1} - 1 \\ &= f_{2(n+1)-1} - 1. \end{aligned}$$

18. We prove this by induction on n. Clearly $\mathbf{A}^1 = \mathbf{A} = \begin{bmatrix} f_2 & f_1 \\ f_1 & f_0 \end{bmatrix}$. Assume the inductive hypothesis. Then

$$\mathbf{A}^{n+1} = \mathbf{A}\mathbf{A}^n = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{bmatrix} = \begin{bmatrix} f_{n+1} + f_n & f_n + f_{n-1} \\ f_{n+1} & f_n \end{bmatrix} = \begin{bmatrix} f_{n+2} & f_{n+1} \\ f_{n+1} & f_n \end{bmatrix};$$

as desired.

20. The max or min of one number is itself; $\max(a_1, a_2) = a_1$ if $a_1 \geq a_2$ and a_2 if $a_1 < a_2$, whereas $\min(a_1, a_2) = a_2$ if $a_1 \geq a_2$ and a_1 if $a_1 < a_2$; and for $n \geq 2$,

$$\max(a_1, a_2, \ldots, a_{n+1}) = \max(\max(a_1, a_2, \ldots, a_n), a_{n+1})$$

and

$$\min(a_1, a_2, \ldots, a_{n+1}) = \min(\min(a_1, a_2, \ldots, a_n), a_{n+1}).$$

22. Clearly only positive integers can be in S, since 1 is a positive integer, and the sum of two positive integers is again a positive integer. To see that all positive integers are in S, we proceed by induction. Obviously $1 \in S$. Assuming that $n \in S$, we get that $n+1$ is in S by applying the recursive part of the definition with $s = n$ and $t = 1$. Thus S is precisely the set of positive integers.

24. a) Odd integers are obtained from other odd integers by adding 2. Thus we can define this set S as follows: $1 \in S$; and if $n \in S$, then $n+2 \in S$.

b) Powers of 3 are obtained from other powers of 3 by multiplying by 3. Thus we can define this set S as follows: $3 \in S$ (this is 3^1, the power of 3 using the smallest positive integer exponent); and if $n \in S$, then $3n \in S$.

c) There are several ways to do this. One that is suggested by Horner's method is as follows. We will assume that the variable for these polynomials is the letter x. All integers are in S (this base case gives us all the constant polynomials); if $p(x) \in S$ and n is any integer, then $xp(x) + n$ is in S. Another method constructs the polynomials term by term. Its base case is to let 0 be in S; and its inductive step is to say that if $p(x) \in S$, c is an integer, and n is a nonnegative integer, then $p(x) + cx^n$ is in S.

26. a) If we apply each of the recursive step rules to the only element given in the basis step, we see that $(2,3)$ and $(3,2)$ are in S. If we apply the recursive step to these we add $(4,6)$, $(5,5)$, and $(6,4)$. The next round gives us $(6,9)$, $(7,8)$, $(8,7)$, and $(9,6)$. A fourth set of applications adds $(8,12)$, $(9,11)$, $(10,10)$, $(11,9)$, and $(12,8)$; and a fifth set of applications adds $(10,15)$, $(11,14)$, $(12,13)$, $(13,12)$, $(14,11)$, and $(15,10)$.

b) Let $P(n)$ be the statement that $5 \mid a+b$ whenever $(a,b) \in S$ is obtained by n applications of the recursive step. For the basis step, $P(0)$ is true, since the only element of S obtained with no applications of the recursive step is $(0,0)$, and indeed $5 \mid 0+0$. Assume the strong inductive hypothesis that $5 \mid a+b$ whenever $(a,b) \in S$ is obtained by k or fewer applications of the recursive step, and consider an element obtained with $k+1$ applications of the recursive step. Since the final application of the recursive step to an element (a,b) must be applied to an element obtained with fewer applications of the recursive step, we know that $5 \mid a+b$. So we just need to check that this inequality implies $5 \mid a+2+b+3$ and $5 \mid a+3+b+2$. But this is clear, since each is equivalent to $5 \mid a+b+5$, and 5 divides both $a+b$ and 5.

c) This holds for the basis step, since $5 \mid 0+0$. If this holds for (a,b), then it also holds for the elements obtained from (a,b) in the recursive step by the same argument as in part (b).

28. a) The simplest elements of S are $(1,2)$ and $(2,1)$. That is the basis step. To get new elements of S from old ones, we need to maintain the parity of the sum, so we either increase the first coordinate by 2, increase the second coordinate by 2, or increase each coordinate by 1. Thus our recursive step is that if $(a,b) \in S$, then $(a+2,b) \in S$, $(a,b+2) \in S$, and $(a+1,b+1) \in S$.

b) The statement here is that b is a multiple of a. One approach is to have an infinite number of base cases to take care of the fact that every element is a multiple of itself. So we have $(n,n) \in S$ for all $n \in \mathbf{Z}^+$. If one objects to having an infinite number of base cases, then we can start with $(1,1) \in S$ and a recursive rule that if $(a,a) \in S$, then $(a+1,a+1) \in S$. Larger multiples of a can be obtained by adding a to a known multiple of a, so our recursive step is that if $(a,b) \in S$, then $(a,a+b) \in S$.

c) The smallest pairs in which the sum of the coordinates is a multiple of 3 are $(1,2)$ and $(2,1)$. So our basis step is $(1,2) \in S$ and $(2,1) \in S$. If we start with a point for which the sum of the coordinates is a multiple of 3 and want to maintain this divisibility condition, then we can add 3 to the first coordinate, or add 3 to the second coordinate, or add 1 to the one of the coordinates and 2 to the other. Thus our recursive step is that if $(a,b) \in S$, then $(a+3,b) \in S$, $(a,b+3) \in S$, $(a+1,b+2) \in S$, and $(a+2,b+1) \in S$.

30. Since we are concerned only with the substrings 01 and 10, all we care about are the changes from 0 to 1 or 1 to 0 as we move from left to right through the string. For example, we view 0011110110100 as a block of

0's followed by a block of 1's followed by a block of 0's followed by a block of 1's followed by a block of 0's followed by a block of 1's followed by a block of 0's. There is one occurrence of 01 or 10 at the start of each block other than the first, and the occurrences alternate between 01 and 10. If the string has an odd number of blocks (or the string is empty), then there will be an equal number of 01's and 10's. If the string has an even number of blocks, then the string will have one more 01 than 10 if the first block is 0's, and one more 10 than 01 if the first block is 1's. (One could also give an inductive proof, based on the length of the string, but a stronger statement is needed: that if the string ends in a 1 then 01 occurs at most one more time than 10, but that if the string ends in a 0, then 01 occurs at most as often as 10.)

32. a) $ones(\lambda) = 0$ and $ones(wx) = x + ones(w)$, where w is a bit string and x is a bit (viewed as an integer when being added)
b) The basis step is when $t = \lambda$, in which case we have $ones(s\lambda) = ones(s) = ones(s)+0 = ones(s)+ones(\lambda)$. For the inductive step, write $t = wx$, where w is a bit string and x is a bit. Then we have $ones(s(wx)) = ones((sw)x) = x + ones(sw)$ by the recursive definition, which is $x + ones(s) + ones(w)$ by the inductive hypothesis, which is $ones(s) + (x + ones(w))$ by commutativity and associativity of addition, which finally equals $ones(s) + ones(wx)$ by the recursive definition.

34. a) 1010 b) 1 1011 c) 1110 1001 0001

36. We induct on w_2. The basis step is $(w_1\lambda)^R = w_1^R = \lambda w_1^R = \lambda^R w_1^R$. For the inductive step, assume that $w_2 = w_3 x$, where w_3 is a string of length one less than the length of w_2, and x is a symbol (the last symbol of w_2). Then we have $(w_1 w_2)^R = (w_1 w_3 x)^R = x(w_1 w_3)^R$ (by the recursive definition given in the solution to Exercise 35). This in turn equals $x w_3^R w_1^R$ by the inductive hypothesis, which is $(w_3 x)^R w_1^R$ (again by the definition). Finally, this equals $w_2^R w_1^R$, as desired.

38. There are two types of palindromes, so we need two base cases, namely λ is a palindrome, and x is a palindrome for every symbol x. The recursive step is that if α is a palindrome and x is a symbol, then $x\alpha x$ is a palindrome.

40. The key fact here is that if a bit string of length greater than 1 has more 0's than 1's, then either it is the concatenation of two such strings, or else it is the concatenation of two such strings with one 1 inserted either before the first, between them, or after the last. This can be proved by looking at the running count of the excess of 0's over 1's as we read the string from left to right. Therefore one recursive definition is that 0 is in the set, and if x and y are in the set, then so are xy, $1xy$, $x1y$, and $xy1$.

42. Recall from Exercise 37 the recursive definition of the i^{th} power of a string. We also will use the result of Exercise 36 and the following lemma: $w^{i+1} = w^i w$ for all $i \geq 0$, which is clear (or can be proved by induction on i, using the associativity of concatenation).

Now to prove that $(w^R)^i = (w^i)^R$, we use induction on i. It is clear for $i = 0$, since $(w^R)^0 = \lambda = \lambda^R = (w^i)^R$. Assuming the inductive hypothesis, we have $(w^R)^{i+1} = w^R(w^R)^i = w^R(w^i)^R = (w^i w)^R = (w^{i+1})^R$, as desired.

44. For the basis step we have the tree consisting of just the root, so there is one leaf and there are no internal vertices, and $l(T) = i(T) + 1$ holds. For the recursive step, assume that this relationship holds for T_1 and T_2, and consider the tree with a new root, whose children are the roots of T_1 and T_2. The new root is an internal vertex of T, and every internal vertex in T_1 or T_2 is an internal vertex of T, so $i(T) = i(T_1) + i(T_2) + 1$. Similarly, the leaves of T_1 and T_2 are the leaves of T, so $l(T) = l(T_1) + l(T_2)$. Thus we have $l(T) = l(T_1) + l(T_2) = i(T_1) + 1 + i(T_2) + 1$ by the inductive hypothesis, which equals $(i(T_1) + i(T_2) + 1) + 1 = i(T) + 1$, as desired.

46. The basis step requires that we show that this formula holds when $(m,n) = (1,1)$. The induction step requires that we show that if the formula holds for all pairs smaller than (m,n) in the lexicographic ordering of $\mathbf{Z}^+ \times \mathbf{Z}^+$, then it also holds for (m,n). For the basis step we have $a_{1,1} = 5 = 2(1+1)+1$. For the inductive step, assume that $a_{m',n'} = 2(m'+n')+1$ whenever (m',n') is less than (m,n) in the lexicographic ordering of $\mathbf{Z}^+ \times \mathbf{Z}^+$. By the recursive definition, if $n = 1$ then $a_{m,n} = a_{m-1,n}+2$; since $(m-1,n)$ is smaller than (m,n), the induction hypothesis tells us that $a_{m-1,n} = 2(m-1+n)+1$, so $a_{m,n} = 2(m-1+n)+1+2 = 2(m+n)+1$, as desired. Now suppose that $n > 1$, so $a_{m,n} = a_{m,n-1}+2$. Again we have $a_{m,n-1} = 2(m+n-1)+1$, so $a_{m,n} = 2(m+n-1)+1+2 = 2(m+n)+1$, and the proof is complete.

48. a) $A(1,0) = 0$ by the second line of the definition.
b) $A(0,1) = 2$ by the first line of the definition.
c) $A(1,1) = 2$ by the third line of the definition.
d) $A(2,2) = A(1,A(2,1)) = A(1,2) = A(0,A(1,1)) = A(0,2) = 4$

50. We prove this by induction on n. It is clear for $n = 1$, since $A(1,1) = 2 = 2^1$. Assume that $A(1,n) = 2^n$. Then $A(1,n+1) = A(0,A(1,n)) = A(0,2^n) = 2 \cdot 2^n = 2^{n+1}$, as desired.

52. This is impossible to compute, if by compute we mean write down a nice numeral for the answer. As explained in the solution to Exercise 51, one can show by induction that $A(2,n)$ is equal to $2^{2^{\cdot^{\cdot^{2}}}}$, with n 2's in the tower. To compute $A(3,4)$ we use the definition to write $A(3,4) = A(2,A(3,3))$. We saw in the solution to Exercise 51, however, that $A(3,3) = 65536$, so $A(3,4) = A(2,65536)$. Thus $A(3,4)$ is a tower of 2's with 65536 2's in the tower. There is no nicer way to write or describe this number—it is too big.

54. We use a double induction here, inducting first on m and then on n. The outside base case is $m = 0$ (with n arbitrary). Then $A(m,n) = 2n$ for all n. Also $A(m+1,n) = 2n$ for $n = 0$ and $n = 1$, and $2n \geq 2n$ in those cases; and $A(m+1,n) = 2^n$ for all $n > 1$ (by Exercise 50), and in those cases $2^n \geq 2n$ as well. Now we assume the inductive hypothesis, that $A(m+1,t) \geq A(m,t)$ for all t. We will show by induction on n that $A(m+2,n) \geq A(m+1,n)$. For $n = 0$ this reduces to $0 \geq 0$, and for $n = 1$ it reduces to $2 \geq 2$. Assume the inner inductive hypothesis, that $A(m+2,n) \geq A(m+1,n)$. Then

$$\begin{aligned}
A(m+2,n+1) &= A(m+1,A(m+2,n)) \\
&\geq A(m+1,A(m+1,n)) \quad \text{(using the inductive hypothesis and Exercise 53)} \\
&\geq A(m,A(m+1,n)) \quad \text{(by the inductive hypothesis on } m\text{)} \\
&= A(m+1,n+1)\,.
\end{aligned}$$

56. Let $P(n)$ be the statement "F is well-defined at n." Then $P(0)$ is true, since $F(0)$ is specified. Assume that $P(n)$ is true. Then F is also well-defined at $n+1$, since $F(n+1)$ is given in terms of $F(n)$. Therefore by mathematical induction, $P(n)$ is true for all n, i.e., F is well-defined as a function on the set of all nonnegative integers.

58. a) This would be a proper definition if the recursive part were stated to hold for $n \geq 2$. As it stands, however, $F(1)$ is ambiguous, and $F(0)$ is undefined.
b) This definition makes no sense as it stands; $F(3)$ is not defined, since $F(0)$ isn't. Also, $F(2)$ is ambiguous.
c) For $n = 3$, the recursive part makes no sense, since we would have to know $F(3/2)$. Also, $F(2)$ is ambiguous.
d) The definition is ambiguous about $n = 1$, since both the second clause and the third clause seem to apply. This would be a valid definition if the third clause applied only to odd $n \geq 3$.

e) We note that $F(1)$ is defined explicitly, $F(2)$ is defined in terms of $F(1)$, $F(4)$ is defined in terms of $F(2)$, and $F(3)$ is defined in terms of $F(8)$, which is defined in terms of $F(4)$. So far, so good. However, let us see what the definition says to do with $F(5)$:

$$F(5) = F(14) = 1 + F(7) = 1 + F(20) = 1 + 1 + F(10) = 1 + 1 + 1 + F(5).$$

This not only leaves us begging the question as to what $F(5)$ is, but is a contradiction, since $0 \neq 3$. (If we replace "$3n - 1$" by "$3n + 1$" in this problem, then it is an unsolved problem—the Collatz conjecture—as to whether F is well-defined; see Example 23 in Section 1.7.)

60. In each case we will apply the definition. Note that $\log^{(1)} n = \log n$ (for $n > 0$). Similarly, $\log^{(2)} n = \log(\log n)$ as long as it is defined (which is when $n > 1$), $\log^{(3)} n = \log(\log(\log n))$ as long as it is defined (which is when $n > 2$), and so on. Normally the parentheses are understood and omitted.

a) $\log^{(2)} 16 = \log\log 16 = \log 4 = 2$, since $2^4 = 16$ and $2^2 = 4$

b) $\log^{(3)} 256 = \log\log\log 256 = \log\log 8 = \log 3 \approx 1.585$

c) $\log^{(3)} 2^{65536} = \log\log\log 2^{65536} = \log\log 65536 = \log 16 = 4$

d) $\log^{(4)} 2^{2^{65536}} = \log\log\log\log 2^{2^{65536}} = \log\log\log 2^{65536} = 4$ by part (c)

62. Note that $\log^{(1)} 2 = 1$, $\log^{(2)} 2^2 = 1$, $\log^{(3)} 2^{2^2} = 1$, $\log^{(4)} 2^{2^{2^2}} = 1$, and so on. In general $\log^{(k)} n = 1$ when n is a tower of k 2s; once n exceeds a tower of k 2s, $\log^{(k)} n > 1$. Therefore the largest n such that $\log^* n = k$ is a tower of k 2s. Here $k = 5$, so the answer is $2^{2^{2^{2^2}}} = 2^{65536}$. This number overflows most calculators. In order to determine the number of decimal digits it has, we recall that the number of decimal digits of a positive integer x is $\lfloor \log_{10} x \rfloor + 1$. Therefore the number of decimal digits of 2^{65536} is $\lfloor \log_{10} 2^{65536} \rfloor + 1 = \lfloor 65536 \log_{10} 2 \rfloor + 1 = 19{,}729$.

64. Each application of the function f divides its argument by 2. Therefore iterating this function k times (which is what $f^{(k)}$ does) has the effect of dividing by 2^k. Therefore $f^{(k)}(n) = n/2^k$. Now $f_1^*(n)$ is the smallest k such that $f^{(k)}(n) \le 1$, that is, $n/2^k \le 1$. Solving this for k easily yields $k \ge \log n$, where logarithm is taken to the base 2. Thus $f_1^*(n) = \lceil \log n \rceil$ (we need to take the ceiling function because k must be an integer).