Problem §3.1: 2: Determine which characteristics of an algorithm described in the text the following procedures have and which they lack.

(a)
```
procedure double(n: positive integer)
while n > 0
    n := 2n
```

(b)
```
procedure divide(n: positive integer)
while n >= 0
    m : = 1/n
    n := n-1
```

(c)
```
procedure sum(n: positive integer)
sum := 0
while i < 10
    sum := sum + i
```

(d)
```
procedure choose(a,b: integers)
x := either a or b
```

Solution. For the sake of concision, we'll just explain the properties that each procedure lacks.

(a) This procedure has every listed characteristic except **finiteness** because the while loop will continue indefinitely. To see this, observe that if the condition for the while loop is met (that $n > 0$), then doubling n will again produce a number greater than zero and the while loop will execute again.

(b) This procedure has every listed characteristic except **effectiveness**. The problem is that the $n := 1/n$ step is not defined when $n = 0$. Because the procedure begins with a positive integer and then uses the while loop to subtract one until n becomes negative, at some point it will reach the $n = 0$ case and encounter this division by zero.

(c) This procedure has every listed characteristic except **definiteness**. The "while $i < 10$" step is not well-defined because the value of i is never set. So a reader trying to execute the procedure would become confused and unable to proceed at that point.

(d) This procedure also has every listed property except **definiteness** because the it doesn't actually tell us whether to set x equal to a or b. So if two people executed this procedure independently, they might end up with different values for x!

□

Problem §3.1: 24: Describe an algorithm that determines whether a function from a finite set to another finite set is one-to-one.

Solution. So we have some notation to work with, let's consider a function $f : A \to B$ where $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_m\}$. The input to the algorithm consists of all $n + m$ elements of A and B and the function f. At the beginning of the algorithm, we'll initialize a list of length m called *hit* which tracks which elements of B are images of elements of A. Until we find an element of A with image b_i, the i^{th} entry of *hit* is 0. Once we find such an element, we set the i^{th} entry to 1. The algorithm runs through the elements of A, computing the image of each one and checking to see if the i^{th} entry of *hit* is 0 or 1. If the entry is already 1, then the function is not one-to-one and the algorithm terminates. If the algorithm is not halted prematurely, then we conclude the function is one-to-one. □

Problem §3.1: 52(a,d): Use the greedy algorithm to make change using quarters, dimes, nickels, and pennies for

(a) 87 cents.

(d) 33 cents.

Solution. Recall that the greedy algorithm makes change by selecting the largest coin whose value does not exceed the amount of change to be given, adding that coin to the pile of change, and then decreasing the amount of change required.

(a) The algorithm first uses the maximum number of quarters possible, 3. This leaves $87-3(25) = 87-75 = 12$ cents remaining. It then uses the maximum possible number of dimes, 1, leaving $12-1(10) = 2$ cents. The algorithm cannot use any nickels, because $5 > 2$. Finally, it uses the maximum possible number of pennies, 2, which brings the amount of change required to 0. The algorithm then terminates.

(d) The algorithm first uses the maximum number of quarters possible, 1, leaving $33-25 = 8$ cents. The algorithm cannot use any dimes, because $10 > 8$. Next, it uses the maximum number of nickels possible, 1, leaving $8-1(5) = 3$ cents. Finally, it uses the maximum number of pennies possible, 3, bringing the amount of change remaining to 0. At this point, the algorithm terminates.

□

Problem §3.1: 54(a,d): Use the greedy algorithm to make change using quarters, dimes, and pennies (but no nickels) for

(a) 87 cents.

(d) 33 cents.

Solution. Now, we run the greedy change algorithm again without nickels.

(a) Again, the algorithm begins by using the maximum possible number of quarters, 3, and leaves $87-3(25) = 12$ cents. It then uses the maximum possible number of dimes, 1, which leaves $12-10 = 2$ cents. Unlike in 52, there are no nickels available so it does not test whether or not it's possible to use a nickel. It then uses the maximum number of pennies, 2, as that is the only remaining coin. This brings the amount of change required to 0 and the algorithm terminates.

Note that we reached the same answer as in 52(a).

(d) Again, the algorithm begins by using the maximum possible number of quarters, 1, which leaves 8 cents. It cannot use any dimes, because $10 > 8$. Because there are no nickels available, it then uses the maximum possible number of pennies, 8, leaving 0 cents. The algorithm then terminates.

Observe that in this case the greedy algorithm required a total of nine coins (one quarter, eight pennies). We could have instead used just six coins by using three dimes and three pennies. As such, this example shows that the greedy change algorithm does not produce an optimal solution for this set of coins.

□

Problem §3.2: 2(a,b,e,f): Determine whether each of these functions is $O(x^2)$:

(a) $f(x) = 17x + 11$.

(b) $f(x) = x^2 + 1000$.

(e) $f(x) = 2^x$.

(f) $f(x) = \lfloor x \rfloor \cdot \lceil x \rceil$.

Solution. (a) **Yes**, $f(x)$ is $O(x^2)$ because

$$|17x + 11| \leq |17x + x| = |18x| \leq |18x^2|$$

for all $x > 11$. The witnesses are $C = 18$ and $k = 11$.

(b) **Yes**, $f(x)$ is $O(x^2)$ because

$$|x^2 + 1000| \leq |x^2 + x^2| = 2x^2$$

for all $x > \sqrt{1000}$. The witnesses are $C = 2$ and $k = \sqrt{1000}$.

(c) **No**, $f(x)$ is not $O(x^2)$. If it were, then we would have $|2^x| < C|x^2|$ for some constant C, but $2^x > x^3$ for all $x \geq 10$. So for large x, $|2^x/x^2| \geq |x^3/x^2| = |x|$ which is certainly not bounded by a constant.

(f) **Yes**, $f(x)$ is $O(x^2)$ because

$$|\lfloor x \rfloor \lceil x \rceil| \leq |x(x+1)| \leq x(2x) = 2x^2$$

for all $x > 1$. The witnesses are $C = 2$ and $k = 1$.

□

Problem §3.2: 8: Find the least integer n such that $f(x)$ is $O(x^n)$ for each of these functions.

(a) $f(x) = 2x^2 + x^3 \log x$.

(b) $f(x) = 3x^5 + (\log x)^4$.

(c) $f(x) = (x^4 + x^2 + 1)/(x^4 + 1)$.

(d) $f(x) = (x^3 + 5\log x)/(x^4 + 1)$.

Solution. For each function, we essentially want to identify its fastest growing term and find an $O(x^n)$ bound for that term.

(a) The fastest growing term is $x^3 \log x$. This term is not $O(x^3)$ because the $\log x$ factor grows without bound as x grows. Because $\log x$ grows more slowly than x, this suggests that it may be $O(x^4)$. To verify this, we observe that

$$|2x^2 + x^3 \log x| \leq |2x^4 + x^4| = 3|x^4|$$

for all $x > 1$. As such, $f(x)$ is $O(x^4)$ with witnesses $C = 3$ and $k = 1$.

(b) The fastest growing term is $3x^5$. To see that $f(x)$ is $O(x^5)$, observe that

$$|3x^5 + (\log x)^4| \leq |3x^5 + x^5| = 4|x^5|$$

for all $x > 1$. As such, $f(x)$ is $O(x^5)$ with witnesses $C = 4$ and $k = 1$.

(c) Informally, we can observe that if we took the limit of $f(x)$ as $x \to \infty$, this function would approach 1. This suggests that the function is $O(1)$. To verify this, observe that

$$\left|\frac{x^4 + x^2 + 1}{x^4 + 1}\right| \leq \left|\frac{x^4 + x^4 + x^4}{x^4 + 1}\right| \leq \left|\frac{x^4 + x^4 + x^4}{x^4}\right| = \left|\frac{3x^4}{x^4}\right| = 3 \cdot |1|,$$

so $f(x)$ is $O(1)$ with witnesses $C = 3$ and $k = 1$.

(d) Again, we can informally observe when x is very large, $f(x) \approx 1/x$. This suggests that the function is $O(1/x)$. To verify this, observe that

$$\begin{aligned} \left|\frac{x^3 + 5\log x}{x^4+1}\right| &\le \left|\frac{x^3+5x^3}{x^4+1}\right| \qquad \text{for } x > 0 \\ &\le \left|\frac{6x^3}{x^4}\right| \\ &= 6\left|\frac{1}{x}\right| \end{aligned}$$

so $f(x)$ is $O(1/x)$ with witnesses $C = 6$ and $k = 0$.

□

Problem §3.2: 17: Suppose that $f(x)$, $g(x)$, and $h(x)$ are functions such that $f(x)$ is $O(g(x))$ and $g(x)$ is $O(h(x))$. Show that $f(x)$ is $O(h(x))$.

Solution. Assume that $f(x)$ is $O(g(x))$, so by definition there exist constants C_1 and k_1 such that $|f(x)| \le C_1|g(x)|$ for $x > k_1$. Assume also that $g(x)$ is $O(h(x))$, so by definition there exist constants C_2 and k_2 such that $|g(x)| \le C_2|h(x)|$ for $x > k_2$. Then observe that

$$|f(x)| \le C_1|g(x)| \le C_1C_2|h(x)|$$

when $x > \max(k_1, k_2)$. Hence, $f(x)$ is by definition $O(h(x))$ with witnesses $C = C_1C_2$ and $k = \max(k_1, k_2)$. □

Problem §3.2: 26: Give a big-O estimate for each of these functions. For the function g in your estimate $f(x)$ is $O(g(x))$, use a simple function g of the smallest order.

(a) $f(x) = (n^3 + n^2\log n)(\log n + 1) + (17\log n + 19)(n^3 + 2)$.

(b) $f(x) = (2^n + n^2)(n^3 + 3^n)$.

(c) $f(x) = (n^n + n2^n + 5^n)(n! + 5^n)$.

Solution. As in Problem 8 from §3.2, we want to identify the fastest growing term. Discarding any constant multiple, this term gives a smallest order big-O estimate for the function.

(a) We look at each term independently. The first term is the product $(n^3 + n^2\log n)(\log n + 1)$. The fastest growing term of the $n^3 + n^2\log n$ factor is n^3 and the fastest growing term of the $\log n + 1$ factor is $\log n$. So this term "grows like" $n^3\log n$. The second term is the product $(17\log n + 19)(n^3 + 2)$. By similar reasoning, this term "grows like" $n^3\log n$. This means the overall funtion is $O(n^3\log n + n^3\log n) = O(2n^3\log n)$. Because constant coefficients aren't important when thinking about big-O estimates, this is equivalent to being $O(n^3\log n)$.

(b) Again, we want to identify the fastest growing term in each factor. The first factor $2^n + n^2$ has fastest growing term 2^n. The second factor $n^3 + 3^n$ has fastest growing term 3^n. As such, $f(x)$ is $O(2^n \cdot 3^n) = O(6^n)$.

(c) The fastest growing term in the first factor, $n^n + n2^n + 5^n$, is n^n. The fastest growing factor in the second term, $n! + 5^n$, is $n!$. As such, $f(x)$ is $O(n^n n!)$.

□

Problem §3.2: 28(a,b,c,d): Determine whether each of the following functions is $\Omega(x)$ and whether it is $\Theta(x)$.

(a) $f(x) = 10$.

(b) $f(x) = 3x + 7$.

(c) $f(x) = x^2 + x + 1$.

(d) $f(x) = 5 \log x$.

Solution. One strategy for finding a big-Theta estimate for for $f(x)$ is, as in previous problems, to look at the fastest growing term.

(a) **No**, this function is not $\Theta(x)$. We know that a function $f(x)$ is $\Theta(x)$ if and only if $f(x)$ is $O(x)$ and x is $O(f(x))$. Clearly $f(x) = 10$ is $O(x)$ because $10 < |x|$ for all $x > 10$. Observe however that x is not $O(10)$ because there are no constants C, k for which $|x| < 10$ for all $x > k$. As such, $f(x) = 10$ is not $\Theta(x)$.

(b) **Yes**, this function is $\Theta(x)$. Observe that $|3x + 7| \leq 4|x|$ for $x > 7$ and that $|3x + 7| \geq 3|x|$ for $x > 0$ (in fact, this is true for all x so we could have said $x > k$ for any choice of k). By definition, this means $f(x) = 3x + 7$ is $O(x)$ and $\Omega(x)$ and is therefore $\Theta(x)$.

(c) **No**, this function is not $\Theta(x)$. The leading term, x^2, grows more quickly than x and therefore $f(x)$ is not $O(x)$ or $\Theta(x)$. (It is, however, $\Omega(x)$!)

(d) **No**, this function is not $\Theta(x)$. The function $\log x$ grows more slowly than x, so $f(x)$ is not $\Omega(x)$ or $\Theta(x)$. (It is, however, $O(x)$!)

□

Problem Extra: Explain what it means for a function to be

(a) $O(1)$.

(b) $\Omega(1)$.

(c) $\Theta(1)$.

Solution. (a) By definition, a function $f(x)$ is $O(1)$ if there exist constants C, k such that $|f(x)| \leq C$ for all $x > k$. In other (more intuitive) words, a function $f(x)$ is $O(1)$ if its absolute value is bounded **above** for all $x > k$.

(b) By definition, a function $f(x)$ is $\Omega(1)$ if there exist constants C, k such that $|f(x)| > C$ for all $x > k$. In other (more intuitive) words, a function $f(x)$ is $\Omega(1)$ if its absolute value is bounded **below** for all $x > k$ so $f(x)$ "stays away" from zero for large enough x. For example, the function $1/x$ is not $\Omega(1)$ because $\lim_{x\to\infty} 1/x = 0$ but the function $x - 5$ is because, for example, $|x - 5| \geq 3$ for $x > 7$.

(c) By definition, a function $f(x)$ is $\Theta(1)$ if there exist positive constants C_1, C_2, and k such that

$$C_1 \leq |f(x)| \leq C_2$$

for all $x > k$. This means that $|f(x)|$ is bounded between two positive constants. So for large x, $|f(x)|$ can't get "too large" or too close to zero.

□

Problem §5.1: 4: Let $P(n)$ be the statement that $1^3 + 2^3 + \cdots + n^3 = (n(n+1)/2)^2$ for the positive integer n.

(a) What is the statement $P(1)$?

(b) Show that $P(1)$ is true, completing the basis step of the proof.

(c) What is the inductive hypothesis?

(d) What do you need to prove in the inductive step?

(e) Complete the inductive step, identifying where you use the inductive hypothesis.

(f) Explain why these steps show that this formula is true whenever n is a positive integer.

Solution. (a) $P(1)$ is the statement

$$1^3 = \left(\frac{1(1+1)}{2}\right)^2.$$

(b) We can easily verify that both sides of $P(1)$ are equal to 1:

$$\left(\frac{1(1+1)}{2}\right)^2 = \left(\frac{2}{2}\right)^2 = 1,$$
$$1^3 = 1.$$

(c) The inductive hypothesis is the statement that

$$1^3 + 2^3 + \cdots + k^3 = \left(\frac{k(k+1)}{2}\right)^2.$$

We denote this statement as $P(k)$.

(d) In the inductive step, we want to show that $P(k)$ implies $P(k+1)$ for each $k \geq 1$. That is, we want to show that by assuming the inductive hypothesis from part (c) we can prove that

$$1^3 + 2^3 + \cdots + k^3 + (k+1)^3 = \left(\frac{(k+1)(k+2)}{2}\right)^2.$$

(e) Beginning with the left hand side of $P(k+1)$, we can observe that

$$\begin{aligned}\left(1^3 + 2^3 + \cdots + k^3\right) + (k+1)^3 &= \left(\frac{k(k+1)}{2}\right)^2 + (k+1)^3 \qquad \text{(by the IHOP)}\\ &= (k+1)^2\left(\frac{k^2}{4} + k + 1\right)\\ &= (k+1)^2\left(\frac{k^2+4k+4}{4}\right)\\ &= \left(\frac{(k+1)(k+2)}{2}\right)^2,\end{aligned}$$

as desired.

(f) Because we've shown that the statement holds for the base case, $n = 1$, and that $P(k)$ implies $P(k+1)$, we know by the principle of mathematical induction that the statement $P(n)$ holds for all positive integers n.

$\square$

Problem §5.1: 6: Prove that $1\cdot 1! + 2\cdot 2! + \cdots + n\cdot n! = (n+1)! - 1$ whenever n is a positive integer.

Solution. We will use mathematical induction to show that

$$1\cdot 1! + 2\cdot 2! + \cdots + n\cdot n! = (n+1)! - 1$$

for all positive integers n.

Base Case: When $n = 1$, we can verify that $1\cdot 1! = 1 = 2! - 1$.

Inductive Step: Assume that $1\cdot 1! + 2\cdot 2! + \cdots + k\cdot k! = (k+1)! - 1$ for $k \in \mathbb{Z}_{>0}$. Observe that

$$\begin{aligned} 1\cdot 1! + 2\cdot 2! + \cdots + k\cdot k! + (k+1)\cdot(k+1)! &= (k+1)! - 1 + (k+1)\cdot(k+1)! && \text{(by the IHOP)} \\ &= (k+2)\cdot(k+1)! - 1. \end{aligned}$$

Conclusion: Because we've shown that the identity holds for $n = 1$ and that $1\cdot 1! + 2\cdot 2! + \cdots + k\cdot k! = (k+1)! - 1$ implies $1\cdot 1! + 2\cdot 2! + \cdots + k\cdot k! + (k+1)\cdot(k+1)! = (k+2)! - 1$ for all $k \in \mathbb{Z}_{>0}$, we conclude by the principle of mathematical induction that the claim holds for all $n \in \mathbb{Z}_{>0}$, as desired. ☐

Problem §5.1: 8: Prove that $2 - 2\cdot 7 + 2\cdot 7^2 - \cdots + 2(-7)^n = (1 - (-7)^{n+1})/4$ whenever n is a nonnegative integer.

Solution. We wish to show that for all $n \in \mathbb{N}$,

$$2 - 2\cdot 7 + 2\cdot 7^2 - \cdots + 2(-7)^n = \frac{1 - (-7)^{n+1}}{4}.$$

Base Case: When $n = 0$, the left-hand side has only the single term 2 and the right-hand side simplifies to

$$\frac{1 - (-7)^{0+1}}{4} = \frac{1 - (-7)}{4} = \frac{8}{4} = 2.$$

Because both sides of the identity are equal to 2, $P(0)$ is true.

Inductive Step: Assume $P(k)$ is true, i.e. that $2 - 2\cdot 7 + 2\cdot 7^2 - \cdots + 2(-7)^k = (1 - (-7)^{k+1})/4$. We can then observe that

$$\begin{aligned} 2 - 2\cdot 7 + 2\cdot 7^2 - \cdots + 2(-7)^k + 2(-7)^{k+1} &= \frac{1 - (-7)^{k+1}}{4} + 2(-7)^{k+1} && \text{(by the IHOP)} \\ &= \frac{1 - (-7)^{k+1} + 8(-7)^{k+1}}{4} \\ &= \frac{1 + 7(-7)^{k+1}}{4} \\ &= \frac{1 - (-7)(-7)^{k+1}}{4} \\ &= \frac{1 - (-7)^{k+2}}{4}. \end{aligned}$$

Conclusion: Because we've shown that the identity holds for $n = 0$ and that $P(k)$ implies $P(k+1)$ for all $k \in \mathbb{N}$, we have therefore shown by mathematical induction that

$$2 - 2\cdot 7 + 2\cdot 7^2 - \cdots + 2(-7)^n = \frac{1 - (-7)^{n+1}}{4}$$

for all $n \in \mathbb{N}$, as desired. ☐