See formatting style explanation on last page if you haven't yet.

Problem 1-8. Prove the correctness of the following algorithm for evaluating a polynomial.

$$\mathrm{P}(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$$

function horner(A, x)
  $p = A_n$
  for i from $n-1$ to 0
    $p = p * x + A_i$
  return p

Solution. This proof is easily done using induction. First, let us verify a base case; we'll chose $n = 1$ as a nice starting point. Note that $\mathrm{P}[1; x] == a_1 x + a_0$. By stepping through a call to horner$[1; A, x]$, we can see with no trouble that p is initialized to A_1. Since $n == 1 \longleftrightarrow n - 1 == 0$ we can see that for loop will execute once, so this sets $p = (A_1)x + A_i = A_1 x + A_0$, which matches the polynomial function.
Now, for the case $n = n+1$, note that $\mathrm{P}[n+1; x] == a_{n+1}x^{n+1} + a_n x^n + \ldots a_1 x + a_0$. Once again, we step through horners. In this case, since we have $n = n + 1$, p is initialized to A_{n+1}. This time around, the for loop runs from $(n + 1) - 1 == n$ to 0, which entails that it will execute its block of operations $n + 1$ times. If we remove the first iteration and re-initialize p to be the value of p after that iteration, we'll have $p == A_{n+1}x + A_n$ and the for loop now runs from $n-1$ to 0 as it originally did. By distributive property of multiplication, it's trivial to conclude that when the loop terminates $p == A_{n+1}x^{n+1} + A_n x^n + \ldots A_1 x + A_0$, which mirrors the output of the polynomial function. $\therefore$ the algorithm is correct.

■

Problem 1-16. Prove by induction that $n^3 + 2n$ is divisible by 3 for all $n \geq 0$.

Solution. First, for my sanity, let's define *function* ThreeGen $[n \in \mathbb{I}]$ { return $n^3 + 2n$; }

Base Case ($n = 1$): By simple substitution ThreeGen$[1] == (1)^3 + 2(1) == 3$, and of course $\forall x \in \mathbb{N}(x \,\%\, x == 0)$ so this base case works.

General Case ($n = n+1$): Once again, we start with substitution: ThreeGen$[n+1] == (n + 1)^3 + 2(n + 1)$. We can further simplify this like so:

$$\text{ThreeGen}[n + 1] == (n + 1)^3 + 2(n + 1)$$
$$\implies (n + 1)^3 + 2n + 2, \text{ Distributive Property of Multiplication}$$
$$\implies (n + 1)(n + 1)(n + 1) + 2n + 2, \text{ Definition of Powers of scalars}$$
$$\implies (n(n + 1) + 1(n + 1))(n + 1) + 2n + 2, \text{ Distributive Property of Multiplication}$$
$$\implies (n^2 + n + n + 1)(n + 1) + 2n + 2, \text{ Distributive Property of Multiplication}$$
$$\implies (n^2 + 2n + 1)(n + 1) + 2n + 2, \text{ Simplify}$$
$$\implies n(n^2 + 2n + 1) + 1(n^2 + 2n + 1) + 2n + 2, \text{ Distributive Property of Multiplication}$$

$$\implies n(n^2) + n(2n) + n(1) + n^2 + 2n + 1 + 2n + 2, \text{ Distributive Property of Multiplication}$$
$$\implies n^3 + 2n^2 + n + n^2 + 4n + 3, \text{ Simplify}$$
$$\implies n^3 + 3n^2 + 5n + 3, \text{ Simplify}$$

Now we can see that the expression is in simplest terms. From here, we'll try to extract something of the form: ThreeGen$[n]$ + $F[n]$ for some function F of n.

$$\text{ThreeGen}[n+1] == n^3 + 3n^2 + 5n + 3$$
$$\implies n^3 + 3n^2 + 3n + 2n + 3, \text{ Definition of Multiplication of scalars}$$
$$\implies n^3 + 2n + 3n^2 + 3n + 3, \text{ Associative Property of Addition}$$
$$\implies n^3 + 2n + 3(n^2 + n + 1), \text{ Distributive Property of Multiplication}$$

We know because of our base case that $(n^3 + 2n)$ % $3 == 0$. Furthermore, by Definition of Multiplication of scalars $\forall x \in \mathbb{N}(3x \ \% \ 3 == 0)$, and by Distribution Property of Multiplication $\forall(x, y) \in \mathbb{N}^2(((x \ \% \ 3 == 0)(y \ \% \ 3 == 0)) \longleftrightarrow ((x + y) \ \% \ 3) == 0)$. Armed with this knowledge, we can see first that the first term $(n^3 + 2n)$ is divisible by 3, then that the second term $(3(n^2 + n + 1)$ is also divisible by 3, and finally that since both terms are divisible by three, so must be their sum. $\therefore$ ThreeGen$[n + 1]$ % $3 == 0$ and thus, by induction, $\forall n \in \mathbb{N}(((n^3 + 2n) \ \% \ 3) == 0)$.

■

Problem 1-25. A sorting algorithm takes 1 second to sort 1,000 items on your local machine. How long will it take to sort 10,000 items...

a. if you believe that the algorithm takes time proportional to n^2, and

b. if you believe that the algorithm takes time roughly proportional to $n \log n$?[1]

Solution.

a. If the algorithm is $O[n^2]$, then multiplying the number of items to sort by 10 increases the time necessary to execute the sort by $10^2 == 100$, so it would take 100 seconds to sort 10,000 items.

b. If the algorithm is $O[n \log[n]]$, then multiplying the number of items to sort by 10 results in an execution time proportional to $10n \log_{10}[10n]$ (in terms of the original n value) which can be expanded to $10n(1 + \log_{10}[n])$. This is related to the original time calculation by a factor of

$$\frac{10}{n \log_1 0[n]} + 10$$

therefore, it takes $10 + 10/(1000 \log_{10}[1000]) == 10 + 10/3000 == 3001/300 == 10.00\overline{3}$ seconds to sort 10,000 items using this algorithm.

■

[1] I'm assuming this is log base 10

Problem 1-29. There are 25 horses. At most, 5 horses can race together at a time. You must determine the fastest, second fastest, and third fastest horses. Find the minimum number of races in which this can be done.

Solution. It can be determined from any single race of 5 horses that two of the participants cannot possibly be in the top three, because the two horses that finish in 4^{th} and 5^{th} are slower than at least 3 other horses. Therefore, by arbitrarily selecting a group of 5, then recursively racing them, selecting the top three to keep and one or two arbitrary new horses (two if possible, otherwise one) until there are no more horses to race, you can reach a conclusion about which three horses are the fastest in only eleven races.

■

Formatting

I've been told that sometimes some of the things I do with regard to to formatting math and logic can be confusing, so typically on a first assignment for a class I accompany the homework with a guide to the style I use, and then the grader can tell me what (if anything) I did that was so confusing I should never do it again. So what follows is a list of stylistic formatting decisions I made in this homework, which is in no particular order, and is by no means exhaustive.

- My function calls and definitions all enclose the argument list with square brackets ('[' and ']') to avoid confusion with grouping symbols. E.g. most people would probably write $\sin(x)$, I would write $\sin[x]$.
- On a somewhat related note, I've picked up a statistician's habit of using both parameters *and* arguments in function calls and definitions as appropriate. For instance, the polynomial function from problem 1-8 is written in the book as $\mathrm{P}(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$, which to me looks like a function that takes a value of x and returns a function of n. Since the *value* of the function depends on x, I call it an argument, while since the entire *form* of the function changes with n, I call that a parameter. The list of parameters is separated from the list of arguments by a semicolon, while within each list items are delimited by commas e.g. "Func[param1, param2; arg1, arg2]".
- Something which is probably familiar to any grader for this particular class is the fact that I use = to denote *assignment* and == to denote *equality.*
- Throughout, I have used % rather than writing out "mod" to indicate a modulus operation, I'm honestly not sure which is standard/preferred.
- Different arrows mean very different things. Logical implication is indicated by "$\longrightarrow$", exclusive nor (if-and-only-if) is indicated by "$\longleftrightarrow$", and "$x \implies y$" means "x produces y" or "x yields y".
- In a derivation, I always start with a line of given logic, usually that something I'm trying to express is equal to some other expression that I'm attempting to restate in a different way. Beneath that, each line takes the form:

  $$\implies \text{expr, Reason}$$

  where "expr" is a mathematical expression, and "Reason" is a law or theorem that allows me to derive this expression from the line that preceded it ("Simplify" indicates just that I've performed actual operations, or that I've combined compatible monomials). The $\implies$ symbol indicates that the expression on this line is another form yielded by the preceding expression, to avoid having to write "Thing == " on every line when it never changes. In general, I tend to be WAY more thorough that any human being ever should be in derivations (yet still somehow botch the simplest algebra), but generally you can just look at the last line to see that I'm headed the right way with the proof.

- Lastly, I have taken Probability and Statistics, Discrete Math, and Digitial Logic, and each of these classes insisted on a different syntax for Boolean algebra. Moving forward with my life, I've decided to go with the one I liked best (Digital Logic, incidentally). So, for two Boolean variables a and b, I write a AND b as ab, a OR b as $a + b$, a XOR b as $a \oplus b$, and NOT a as $\overline{a}$

That should do it. FYI it's my policy to try to replicate the question's formatting perfectly, unless there's a typo or something, so all of that should be just as it looks in the book. If anything else gives you pause, I'm usually pretty good about answering my email these days.