Algorithm for Evaluating x^n

Given a real number x and a positive integer n, this algorithm computes $P = x^n$.

Step 1 (initialization)
Set $P = x$ and $k = 1$.

Step 2 (next power)
while $k < n$
  (a) Replace P with Px.
  (b) Replace k with $k + 1$.
endwhile

Step 3 (output $P = x^n$)
Print P.

Polynomial Evaluation Algorithm

This algorithm computes $P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$, given the nonnegative integer n and real numbers x, $a_0, a_1, \ldots, a_n$.

Step 1 (initialization)
Set $S = a_0$ and $k = 1$.

Step 2 (add next term)
while $k \le n$
  (a) Replace S with $S + a_k x^k$.
  (b) Replace k with $k + 1$.
endwhile

Step 3 (output $P(x) = S$)
Print S.

Horner's Polynomial Evaluation Algorithm

This algorithm computes $P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$, given the nonnegative integer n and real numbers x, $a_0, a_1, \ldots, a_n$.

Step 1 (initialization) Set $S = a_n$ and $k = 1$.

Step 2 (compute next expression)
while $k \le n$
  (a) Replace S with $xS + a_{n-k}$.
  (b) Replace k with $k + 1$.
endwhile

Step 3 (output $P(x) = S$)
Print S.

Next Subset Algorithm

Given a positive integer n and the string $a_1a_2\ldots a_n$ of 0s and 1s corresponding to a subset of a set with n elements, this algorithm computes the string corresponding to the next subset.

Step 1 (initialization)
Set $k = n$.
Step 2 (look for rightmost 0)
while $k \geq 1$ and $a_k = 1$
Replace k with $k - 1$.
endwhile
Step 3 (if there is a zero, form the next string)
if $k \geq 1$
Step 3.1 (change the rightmost 0 to 1)
Replace a_k with 1.
Step 3.2 (change succeeding 1s to 0s)
for $j = k + 1$ to n
Replace a_j with 0.
endfor
Step 3.3 (output)
Print $a_1a_2\ldots a_n$.
otherwise
Step 3.4 (no successor)
Print "This string contains all 1s."
endif

Bubble Sort Algorithm

This algorithm places the numbers in the list $a_1, a_2, \ldots, a_n$ in nondecreasing order.

Step 1 (set beginning of sublist)
for $j = 1$ to $n - 1$
Step 1.1 (find smallest element of sublist)
for $k = n - 1$ to j by -1
Step 1.1.1 (interchange if necessary)
if $a_{k+1} < a_k$
Interchange the values of a_k and a_{k+1}.
endif
endfor
endfor
Step 2 (output list in nondecreasing order)
Print $a_1, a_2, \ldots, a_n$.

EXERCISES 1.4

In Exercises 1–6, tell whether the given expression is a polynomial in x or not, and if so, give its degree.

1. $5x^2 - 3x + \frac{1}{2}$ **2.** 16 **3.** $x^3 - \frac{1}{x^2}$

4. $2^x + 3x$ **5.** $\frac{1}{2x^2 + 7x + 1}$ **6.** $2x + 3x^{1/2} + 4$

all ?

In Exercises 7–10, compute the various values S takes on when the polynomial evaluation algorithm is used to compute P(x). Then do the same thing, using Horner's polynomial evaluation algorithm.

7. $P(x) = 5x + 3,\ x = 2$ **8.** $P(x) = 3x^2 + 2x - 1,\ x = 5$

9. $P(x) = -x^3 + 2x^2 + 5x - 7,\ x = 2$ **10.** $P(x) = 2x^3 + 5x^2 - 4,\ x = 3$

In Exercises 11–14, tell what next string will be produced by the next subset algorithm.

11. 110101 **12.** 110111 **13.** 001101 **14.** 001001

In Exercises 15–18, make a table listing the values of k, j, and $a_1, a_2, \ldots, a_n$ after each step when the next subset algorithm is applied to the given string.

15. 101 **16.** 111 **17.** 1101 **18.** 1110

In Exercises 19–22, illustrate as in Example 1.5 the use of the bubble sort algorithm to sort each given list of numbers.

19. 13, 56, 87, 42 **20.** 23, 5, 17, 12 **21.** 6, 33, 20, 200, 9 **22.** 88, 2, 75, 10, 48

In Exercises 23–26, estimate how long a computer doing one million operations per second would take to do 3^n and $100n^3$ operations.

23. $n = 20$ **24.** $n = 30$ **25.** $n = 40$ **26.** $n = 50$

In Exercises 27–30, tell how many elementary operations the given algorithm uses. (It depends on n.)

27. Algorithm for evaluating $n!$.

Step 1 Set $k = 0$ and $P = 1$.
Step 2 **while** $k < n$
 (a) Replace k with $k + 1$.
 (b) Replace P with kP.
 endwhile
Step 3 Print P.

28. Algorithm for computing the sum of an arithmetic progression of n terms with first term a and common difference d.

Step 1 Set $S = a$, $k = 1$, and $t = a$.
Step 2 **while** $k < n$
 (a) Replace t with $t + d$.
 (b) Replace S with $S + t$.
 (c) Replace k with $k + 1$.
 endwhile
Step 3 Print S.

---

29. Algorithm for computing the sum of a geometric progression of n terms with first term a and common ratio r.

Step 1 Set $S = a$, $P = ar$, and $k = 1$.
Step 2 **while** $k < n$
(a) Replace S with $S + P$.
(b) Replace P with Pr.
(c) Replace k with $k + 1$.
endwhile
Step 3 Print S.

---

30. Algorithm for computing F_n, the nth Fibonacci number (defined in Section 2.5).

Step 1 Set $a = 1$, $b = 1$, $c = 2$, and $k = 1$.
Step 2 **while** $k < n$
(a) Replace c with $a + b$.
(b) Replace a with b.
(c) Replace b with c.
(d) Replace k with $k + 1$.
endwhile
Step 3 Print b.

The polynomial evaluation algorithm is inefficient because it computes x^k anew for each value of k. The following revision corrects this shortcoming.

Revised Polynomial Evaluation Algorithm

This algorithm computes $P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$, given the nonnegative integer n and real numbers $x, a_0, a_1, \ldots, a_n$.

Step 1 (initialization) Set $S = a_0$, $y = 1$, and $k = 1$.
Step 2 (add next term)
while $k \le n$
(a) Replace y with xy.
(b) Replace S with $S + ya_k$.
(c) Replace k with $k + 1$.
endwhile
Step 3 (output $P(x) = S$)
Print S.

In Exercises 31–32, compute the various values S takes on when the revised polynomial evaluation algorithm is used to compute $P(x)$, where $P(x)$ and x are as in the given exercise.

31. Exercise 9

32. Exercise 10

33. Show that the complexity of the revised polynomial evaluation algorithm is $5n + 1$.