

ERRORS IN HACKER'S DELIGHT

Third printing

P. 2: The latest C standard allows `//`-style comments. Therefore, replace the entire second paragraph with “Programs too long or complex for computer algebra are written in the C programming language, as defined by the ISO 1999 standard.”

—Derek M. Jones

P. 4 line 14: Append a semicolon in the “code” font, so that the line is

```
do statement while (expression) ;
```

P. 15 line 4: Add after “formulas.”, “(The first two methods can fail on a machine that has modulo 32 shifts.)”

—Ray Roth

P. 18 lines 17-18: Replace the sentence “Assuming the machine has mod 64 shifts, the first four formulas hold for $0 \leq n \leq 31$, and the last holds for $0 \leq n \leq 63$.” to “These formulas hold for $0 \leq n \leq 31$ and, if the machine has mod 64 shifts, the last holds for $0 \leq n \leq 63$.”

P. 40 paragraphs 4 and 5: One cycle count and three instructions counts are incorrect. The material, starting from the fourth text line of the fourth paragraph, should read:

1’s in fields A , C , and E . This code requires 11 instructions and six cycles on a machine with unlimited instruction-level parallelism, allowing for four instructions to load the two masks.

A method that requires only eight instructions

P. 68 line 10: “returns” should be “return”.

P. 68 line 17: The two occurrences of b should be $b - 1$, so that the sentence is “... mod $b - 1$ to the sum of the digits and, of course, is less than $b - 1$.”

—Svend Jacobsen

P. 68 line 24: “13 instructions” should be “15 instructions”.

P. 69 line 7: “This is 19 instructions” should be “This is 20 instructions”.

P. 75: The last line of the second paragraph should read “own inverse, $y_i \oplus y_j$ is the parity of bits $i - 1$ through j of x , for $i \geq j$.”

—Michael Pyne

P. 75, second from last paragraph, add the sentence: “This is called the “parallel suffix” operation, because each bit is a function of itself and the bits that follow it.”

P. 82: This is a clarification, not a correction of an error. Append to the end of the third paragraph (after “... blocking the carry.”), “If k is a 64-bit quantity, this correction is also needed for the code of Figure 5–12 and for the first of the three variations given above.”

—Donald E. Knuth

The changes below, to pp 117–122, change “parallel prefix” to “parallel suffix,” essentially.

—Donald E. Knuth

P. 117 paragraph 3: Change “parallel prefix” to “parallel suffix” in two places, and change “PP-XOR” to “PS-XOR”.

P. 118 paragraph 2: Change “PP-XOR” to “PS-XOR” in two places.

P. 119 paragraph 1: Change “PP-XOR” to “PS-XOR. In Figure 7–6, change “// Parallel prefix.” to “// Parallel suffix.” And in the title of this figure, change “prefix” to “suffix”.

P. 120: In Figure 7–7, change the five occurrences of “After PP” to “After PS”. Also, in the title, change “prefix” to “suffix”.

P. 121, at the very end of paragraph 2: Change “Figure 7–7” to “Figure 7–6”.

P. 122 paragraph 3: Change “prefix” to “suffix” in two places.

P. 139, there are two places where “ \neq ” should be “ $>$ ”. In Theorem D3 and its corollary, change “integer $\neq 0$ ” to “integer > 0 ”.

—Svend Jacobsen

P. 141 line 25: The comment should be `// 0 <= s <= 15`.

P. 151 line 26: “`u32`” should be “`un32`”.

P. 171 line 15: In the formula for n_c , one of the plus signs should be a minus sign. The first alternative should be “ $2^{W-1} - \text{rem}(2^{W-1}, d) - 1$, if $d > 0$,”. (The second alternative is ok.)

—Svend Jacobsen

P. 183, second displayed formula, change the second \leq to $<$, so that it becomes

$$2 \leq m < 2^W + 2.$$

—Jeff Gros

P. 186 line 4: Change “less than 100” to “less than or equal to 100”.

P. 195 line 19: Change the phrase “in the domain of modular arithmetic on integers” to “modulo any power of 2” (retain the exclamation point).

P. 197, after the second paragraph (just before the section “Sample Multiplicative Inverses), insert the following paragraph:

The “Newton method” described here applies only when (1) the modulus is an integral power of some number a , and (2) the multiplicative inverse of d modulo a is known and is used for the starting point x_0 . It works particularly well for $a = 2$, because then the multiplicative inverse of any (odd) number d modulo 2 is known immediately—it is 1.

P. 207 line 10 in Figure 11–2: “ $x = 5$ ” should be “ $s = 5$ ”.

—Bob Bradley

P. 212 Figure 11–5, replace the entire code with the code shown below. This is necessary to make the remarks about a 64-bit version be correct.

```
int icbrt(unsigned x) {
    int s;
    unsigned y, b;

    y = 0;
    for (s = 30; s >= 0; s = s - 3) {
        y = 2*y;
        b = (3*y*(y + 1) + 1) << s;
        if (x >= b) {
            x = x - b;
            y = y + 1;
        }
    }
    return y;
}
```

P. 215: Delete the fourth bullet and its displayed formula (“It preserves the mathematical identity...”, as it doesn’t make a lot of sense.

—Donald E. Knuth

P. 284 line 9: After “e.g.,” insert “assuming the Riemann Hypothesis,”. And in line 10, change “integer ≥ 3 ” to “real number ≥ 2.106 ”.

P. 303: Change the first index heading under P from “parallel prefix operation” to “parallel prefix and suffix operations.”

P. 305: Just before the index entry for space-filling curves, insert the entry “snoob, 14”.

Non-substantive errors (typographical etc.)

P. 4 lines 8, 14, and 18: The left and right parentheses, and the semicolons in line 18, should be in the “code” font (same font as the words `while` and `for`).

P. 68 line 13: The period after “multiple of 3)” should be before the parenthesis.

P. 84, first formula: The “1” should be in boldface.

P. 101 paragraph 3: Replace the phrase “A small improvement results on most machines ...” with “A small improvement may result on some machines”

P. 266, 5th line below the figure: change “inverse” to “reciprocal”.