

## Errata: *A Computational Introduction to Number Theory and Algebra (Version 2)*

Last updated: 1/22/2012

- p. 45:** last line of Exercise 2.41. “ $2p^f$ ” should be “ $2p^{f/2}$ ”. [Chihong Joo, 9/9/2010]
- p. 45:** Exercise 2.44. First line, replace “ $\equiv 0 \pmod{n}$ ” with “ $= n$ ”; last line, replace “ $\equiv 0 \pmod{n/p^2}$ ” with “ $= n/p^2$ ”. [Chihong Joo, 9/9/2010]
- p. 60:** line 14 of Fig. 3.1. “ $b_i$ ” should be “ $b_j$ ”. Note that this typo was already present in Version 1. [Christophe Weis, 12/28/2008]
- p. 63:** Exercise 3.24. The statement is correct, but can be improved: the inequality “ $\text{len}(a) - \text{len}(b) - 1 \leq \text{len}(q) \leq \text{len}(a) - \text{len}(b) + 1$ ” may be replaced by the inequality “ $\text{len}(a) - \text{len}(b) \leq \text{len}(q) \leq \text{len}(a) - \text{len}(b) + 1$ ”. [Thai Duong, 7/24/2009]
- p. 64:** Exercise 3.25. Again, the statement is correct, but can be improved: the value  $\sum_{i=1}^k \text{len}(n_i) - k$  can be replaced by  $\sum_{i=1}^k \text{len}(n_i) - k + 1$ . [Thai Duong, 7/27/2009]
- p. 142:** Exercise 6.17. “ $n/\text{gcd}(m, n)$ ” should be “ $\text{gcd}(m, n)$ ”. [Thai Duong, 6/23/2009]
- p. 171:** Theorem 7.3, first line of proof. “ $ab = bc$  implies” should be “ $ab = ac$ ”. [Hrvoje Bandov, 8/8/2010]
- p. 200:** Line 7. “Example 7.45” should be “Example 7.54”. [Michael Forbes, 1/22/2012]
- p. 389:** line 18. “affected” should be “effected”. [VS, 11/6/2009]
- p. 390:** line 4 of Fig. 14.1. “ $i \leq m$ ” should be “ $i < m$ ”. [Artem Pelenitsyn, 12/7/2010]
- p. 485:** footnote. The paper by Umans has appeared in pages 481–490 of the STOC proceedings. The paper by Kedlaya and Umans appeared in pages 146–155 of *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2008. A draft of a merged version of these two papers appears on Umans’ web site: [www.cs.caltech.edu/~umans/research](http://www.cs.caltech.edu/~umans/research).

Their algorithms do not treat the ring as an abstract data type, but rather, work directly on concrete representations of certain finite rings (including most finite fields of practical interest). Using fast algorithms for polynomial and integer arithmetic, their algorithm solves the modular composition problem over a finite field of order  $q$  in time (i.e., bit complexity)  $O(\ell^{1+o(1)} \cdot \text{len}(q)^{1+o(1)})$ . The resulting algorithm for computing minimal polynomials over a finite field of order  $q$  (see footnote, p. 508) runs in time  $O(\ell^{1+o(1)} \cdot \text{len}(q)^{1+o(1)})$ . The resulting polynomial

factorization algorithm (see footnote, p. 547) factors a polynomial of degree  $\ell$  over a finite field of order  $q$  in time

$$O\left(\ell^{1.5+o(1)} + \ell^{1+o(1)} \text{len}(q) \cdot \text{len}(q)^{1+o(1)}\right).$$

[VS, 1/15/2009]