We collect corrections and remarks for the LEDA book. In each case we give the location, the person who pointed out the correction or made the remark, and the change or addition to be made.

## Chapter 1, Introduction

**Section 1.4, page 11:** submitted by authors on January 15th, 2000.
*Replace*
Stefan Reinelt
*by*
Gerhard Reinelt

**Section 1.5, Table 1.2, page 15:** submitted by Andrew Goldberg on March 15, 2000.
LEDA and CG codes reported on in Table 1.2 use different initializations that cause very different behavior on pathological problems. Andrew Goldberg has revised his code. The only difference in the new CG code is the initialization, which is as suggested in [GT88] and compatible with that of LEDA. The running time of the CG implementation on AMO and CG2 improved drastically, and the running time on other generators stayed about the same. The running times of the new code can be found below.

**Section 1.5, Table 1.2, page 15:** submitted by authors on March 15, 2000.
We generated an updated version of Table 1.2 using Goldberg's new code. The table was generated on the same machine as the original tables, but with newer compiler versions. The running time of the CG-implementation on AMO improved dramatically, the running time on the other generators stayed about the same.

| Type of network | LEDA | CG |
|---|---|---|
| random (4000 nodes 28000 edges) | 0.24 | 0.15 |
| CG1 (8002 nodes 12003 edges) | 6.74 | 4.23 |
| CG2 (8002 nodes 12001 edges) | 0.07 | 0.69 |
| AMO (4001 nodes 7998 edges) | 0.03 | 0.01 |

**Table 0.1** An updated version of Table 1.2, page 15.

**Chapter 6, Graphs and Their Data Structures**

**Section 6.5, page 252:** submitted by authors on February 25, 2000.
*Replace*
It can be contained in several node lists, but in each particular node list it can appear only once.
*by*
In the current LEDA version (4.1) a node can be contained in only one (!!) *node_list* and in this list it can appear only once.

**Section 6.5, page 252:** submitted by authors on February 25, 2000.
*Replace*
*snode_list*
*by*
*node_slist*

**Section 6.8, page 262, line -10:** submitted by Jop Sibeyn on January 7, 2000.
*Replace*
In particular, if the third face is empty the running time is $O((n+m)\alpha(n))$.
*by*
In particular, if the third face is empty the running time is $O(n \log n + (n+m)\alpha(n))$".

**Section 6.5, page 252, lines 7 and 9 (without counting the figure):** submitted by Gabriel Valiente on January 26, 2000.
*Replace*
member ship
*by*
membership

**Section 6.12, page 275:** submitted by Gabriel Valiente on January 26, 2000.
Edge label $h$ is missing in graph (a)

**Section 6.12, page 276:** submitted by Gabriel Valiente on January 26, 2000.
*Replace*
if the $G$ is acyclic
*by*
if $G$ is acyclic

## 0.1 Chapter 7, Graph Algorithms

**Section 7.5.6, page 333, line -5:** submitted by Uli Meyer on January 28, 2000.
*Replace*

$O(m + m \log n)$
*by*
$O(m + n \log n)$

**Section 7.10, page 445, Figure 7.39:** submitted by Noburu Matsuda on March 2, 2000. The edges incident to the bottom-most node point into the wrong direction. Their direction needs to be reversed.

**Section 7.10.2, page 452, Proof of Lemma 37:** submitted by Noburu Matsuda on March 2, 2000.
*Replace*
We conclude $\sum_{u \in S} excess(v) \leq 0$
*by*
We conclude $\sum_{u \in S} excess(u) \leq 0$

## Chapter 8, Embedded Graphs

**Section 8.5, page 515, line 3:** submitted by Gabriel Valiente on January 26, 2000.
*Replace*
$f = 2$
*by*
$fc = 2$

## Chapter 10, Geometry Algorithms

**Section 10.6, page 715, line 8:** submitted by Michael Seel on March 15, 2000. triangulations!walk through a triangulation is not part of the text, but the argument of an index command

**Section 10.7, page 742, lines $-8$ and $-7$:** submitted by Michael Seel on March 15, 2000.
*Replace*
```
if ( identical(p_sweep,s1.source()) ) s = orientation(s2,p_sweep);
else if ( identical(p_sweep,s2.source() ) s = orientation(s1,p_sweep);
```

*by*
```
if ( identical(p_sweep,s1.source()) ) s = orientation(s2,p_sweep);
else if ( identical(p_sweep,s2.source() ) s = - orientation(s1,p_sweep);
```
The second line is never reached in the LEDA code and hence the error does not affect the segment intersection code.

### Chapter 13, On the Implementation of LEDA

**page 929, line 11 from bottom:** submitted by authors on February 29, 2000.
*Replace*
constuct
*by*
construct