Create account Log in

Article Talk

Read Edit View history

Search

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Add links

# Damm algorithm

From Wikipedia, the free encyclopedia

In error detection, the **Damm algorithm** is a check digit algorithm that detects all single-digit errors and all adjacent transposition errors. It was presented by H. Michael Damm in 2004.[1]

**Contents** [hide]

## Strengths and weaknesses [edit]

The Damm algorithm is similar to the Verhoeff algorithm. It too will detect *all* occurrences of the two most frequently appearing types of transcription errors, namely altering one single digit, and transposing two adjacent digits (including the transposition of the trailing check digit and the preceding digit).[1][2] But the Damm algorithm has the benefit that it makes do without the dedicatedly constructed permutations and its position specific powers being inherent in the Verhoeff scheme. Furthermore, a table of inverses can be dispensed with provided all main diagonal entries of the operation table are zero.

The Damm algorithm does not suffer from exceeding the number of 10 possible values, resulting in the need for using a non-digit character (as the X in the 10-digit ISBN check digit scheme).

Prepending leading zeros does not affect the check digit.[1]

There are totally anti-symmetric quasigroups that detect all phonetic errors associated with the English language (13 ↔ 30, 14 ↔ 40, ..., 19 ↔ 90). The table used in the illustrating example is based on an instance of such kind.

Despite its desirable properties in typical contexts where similar algorithms are used, the Damm algorithm is largely unknown and scarcely used in practice.

## Design [edit]

Its essential part is a quasigroup of order 10 (i.e. having a 10 × 10 Latin square as the body of its operation table) with the special feature of being weakly totally anti-symmetric.[3][4][i][ii][iii] Damm revealed several methods to create totally anti-symmetric quasigroups of order 10 and gave some examples in his doctoral dissertation.[3][i] With this, Damm also disproved an old conjecture that totally anti-symmetric quasigroups of order 10 do not exist.[5]

A quasigroup $(Q, *)$ is called totally anti-symmetric if for all $c, x, y \in Q$, the following implications hold:[4]

1. $(c * x) * y = (c * y) * x \Rightarrow x = y$
2. $x * y = y * x \Rightarrow x = y$,

and it is called weak totally anti-symmetric if only the first implication holds. Damm proved that the existence of a totally anti-symmetric quasigroup of order $n$ is equivalent to the existence of a weak totally anti-symmetric quasigroup of order $n$. For the Damm algorithm with the check equation $(\ldots((0 * x_m) * x_{m-1}) * \ldots) * x_0 = 0$ a weak totally anti-symmetric quasigroup with the property $x * x = 0$ is needed. Such a quasigroup can be constructed from any totally anti-symmetric quasigroup by rearranging the columns in such a way that all zeros lay on the diagonal. And, on the other hand, from any weak totally anti-symmetric quasigroup a totally anti-symmetric quasigroup can be constructed by rearranging the columns in such a way that the first row is in natural order.[3]

## Algorithm [edit]

The validity of a digit sequence containing a check digit is defined over a quasigroup. A quasigroup table ready for use can be taken from Damm's dissertation (pages 98, 106, 111).[3] It is useful if each main diagonal entry is 0,[1] because it simplifies the check digit calculation.

### Validating a number against the included check digit [edit]

1. Set up an interim digit and initialize it to 0.
2. Process the number digit by digit: Use the number's digit as column index and the interim digit as row index, take the table entry and replace the interim digit with it.
3. The number is valid if and only if the resulting interim digit has the value of 0.[1]

### Calculating the check digit [edit]

**Prerequisite:** The main diagonal entries of the table are 0.

1. Set up an interim digit and initialize it to 0.
2. Process the number digit by digit: Use the number's digit as column index and the interim digit as row index, take the table entry and replace the interim digit with it.
3. The resulting interim digit gives the check digit and will be appended as trailing digit to the number.[1]

## Example [edit]

There will be used the operation table set out below.[1] It may be obtained from the totally anti-symmetric quasigroup in Damm's doctoral dissertation page 111[3] by rearranging the rows and changing the entries correspondingly.

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | **0** | **3** | **1** | **7** | **5** | **9** | **8** | **6** | **4** | **2** |
| 1 | **7** | **0** | **9** | **2** | **1** | **5** | **4** | **8** | **6** | **3** |
| 2 | **4** | **2** | **0** | **6** | **8** | **7** | **1** | **3** | **5** | **9** |
| 3 | **1** | **7** | **5** | **0** | **9** | **8** | **3** | **4** | **2** | **6** |
| 4 | **6** | **1** | **2** | **3** | **0** | **4** | **5** | **9** | **7** | **8** |
| 5 | **3** | **6** | **7** | **4** | **2** | **0** | **9** | **5** | **8** | **1** |
| 6 | **5** | **8** | **6** | **9** | **7** | **2** | **0** | **1** | **3** | **4** |
| 7 | **8** | **9** | **4** | **5** | **3** | **6** | **2** | **0** | **1** | **7** |
| 8 | **9** | **4** | **3** | **8** | **6** | **1** | **7** | **2** | **0** | **5** |
| 9 | **2** | **5** | **8** | **1** | **4** | **3** | **6** | **7** | **9** | **0** |

Suppose we choose the number (digit sequence) **572**.

### Calculating the check digit [edit]

| | | | |
|---|---|---|---|
| **digit to be processed → column index** | 5 | 7 | 2 |
| **old interim digit → row index** | **0** | 9 | 7 |
| **table entry → new interim digit** | 9 | 7 | **4** |

The resulting interim digit is **4**. This is the calculated check digit. We append it to the number and obtain **5724**.

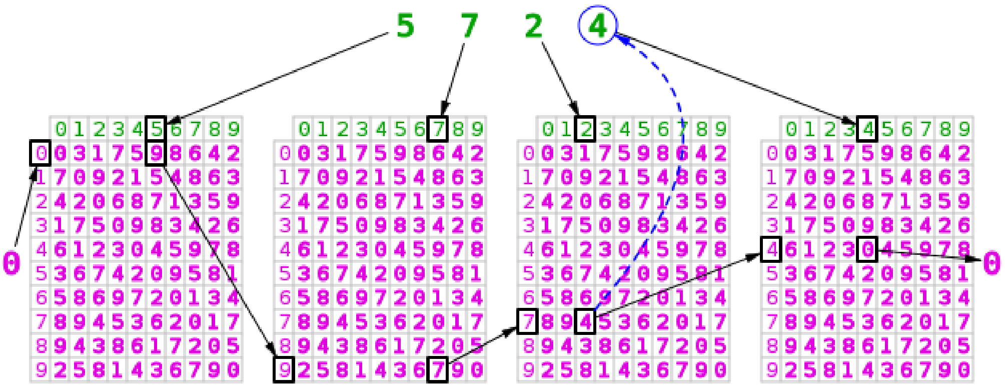
### Validating a number against the included check digit [edit]

| | | | | |
|---|---|---|---|---|
| **digit to be processed → column index** | 5 | 7 | 2 | 4 |
| **old interim digit → row index** | **0** | 9 | 7 | 4 |
| **table entry → new interim digit** | 9 | 7 | 4 | **0** |

The resulting interim digit is **0**, hence the number is **valid**.

### Graphical illustration [edit]

This is the above example showing the detail of the algorithm generating the check digit (broken blue arrow) and verifying the number **572** with the check digit.



|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 3 | 1 | 7 | 5 | 9 | 8 | 6 | 4 | 2 |
| 1 | 7 | 0 | 9 | 2 | 1 | 5 | 4 | 8 | 6 | 3 |
| 2 | 4 | 2 | 0 | 6 | 8 | 7 | 1 | 3 | 5 | 9 |
| 3 | 1 | 7 | 5 | 0 | 9 | 8 | 3 | 4 | 2 | 6 |
| 4 | 6 | 1 | 2 | 3 | 0 | 4 | 5 | 9 | 7 | 8 |
| 5 | 3 | 6 | 7 | 4 | 2 | 0 | 9 | 5 | 8 | 1 |
| 6 | 5 | 8 | 6 | 9 | 7 | 2 | 0 | 1 | 3 | 4 |
| 7 | 8 | 9 | 4 | 5 | 3 | 6 | 2 | 0 | 1 | 7 |
| 8 | 9 | 4 | 3 | 8 | 6 | 1 | 7 | 2 | 0 | 5 |
| 9 | 2 | 5 | 8 | 1 | 4 | 3 | 6 | 7 | 9 | 0 |

## References [edit]

1. ^ a b c d e f g Fenwick, Peter (2014). "Checksums and Error Control". *Introduction to Computer Data Representation*. Bentham Science Publishers. pp. 191–218. doi:10.2174/9781608058822114010013. ISBN 978-1-60805-883-9.
2. ^ For the types of common errors and their frequencies, see Salomon, David (2005). *Coding for Data and Computer Communications*. Springer Science+Business Media, Inc. p. 36. ISBN 978-0387-21245-6.
3. ^ a b c d e Damm, H. Michael (2004). *Total anti-symmetrische Quasigruppen* (PDF) (Dr. rer. nat.) (in German). Philipps-Universität Marburg. urn:nbn:de:hebis:04-z2004-05162.
4. ^ a b Damm, H. Michael (2007). "Totally anti-symmetric quasigroups for all orders *n*≠2,6". *Discrete Mathematics* **307** (6): 715–729. doi:10.1016/j.disc.2006.05.033. ISSN 0012-365X.
5. ^ Damm, H. Michael (2003). "On the Existence of Totally Anti-Symmetric Quasigroups of Order 4*k* + 2". *Computing* **70** (4): 349–357. doi:10.1007/s00607-003-0017-3. ISSN 0010-485X.

i. ^ a b Beliavscaia Galina; Izbaş Vladimir; Şcerbacov Victor (2003). "Check character systems over quasigroups and loops" (PDF). *Quasigroups and Related Systems* **10** (1): 1–28. ISSN 1561-2848. See page 23.

ii. ^ Chen Jiannan (2009). "The NP-completeness of Completing Partial anti-symmetric Latin squares" (PDF). *Proceedings of 2009 International Workshop on Information Security and Application (IWISA 2009)*. Academy Publisher. pp. 322–324. ISBN 978-952-5726-06-0. See page 324.

iii. ^ Mileva, A.; Dimitrova, V. (2009). "Quasigroups constructed from complete mappings of a group ($Z_2^n$, ⊕)" (PDF). *Contributions, Sec. Math. Tech. Sci., MANU/MASA* (Skopje: Macedonian Academy of Sciences and Arts) **XXX** (1-2): 75–93. ISSN 0351-3246. See page 78.

## External links [edit]

- Damm validation & generation code in several programming languages
- Practical application in Singapore
- Quasigroups for the Damm algorithm up to order 64

Wikibooks has a book on the topic of: ***Algorithm Implementation/Checksums/Damm Algorithm***

Categories: Checksum algorithms | Algebraic structures | Latin squares | Group theory