Real World Algorithms: A Beginners Guide
Errata to First Printing

Last updated 14 December 2017

This document lists the changes that should be made to *Real World Algorithms* to correct mistakes that made their way to printing, to improve infelicities that the author spotted too late, or update the material with something that the author did not know at the time of writing the book.
There are three different kinds of changes noted here. In all of them the date that they became known to the author is given at the first line of each item. The name of the person who suggested the change is also given at the end of each change.

► **Page 1** line 1 — 1 Jan 1

These are technical or typographical errors.

Page 1 line 1 — 1 Jan 1

These as changes that improve the book, even if they do not correct an error. They include small rewordings, or material that became known to the author after the book was published.

Page 1 line 1 — 1 Jan 1

These are minor fixes that although they do not make a big difference they do hurt the author. Some of them might strain the reader's eye to see where the improvement is exactly.

► **Page xii** line 2 — 24 Apr 2017

they can proved ↝ they can be proved (S. Subramanya)

Page 8 line −8 and −2 — 12 Aug 2017

big-Oh ↝ big O

Page 9 line 4 — 12 Aug 2017

big-Ohs ↝ big Os

Page 9 line −11 — 12 Aug 2017

In terms of big-Oh notation, we have by definition that ↝ In terms of big O notation, we have, by definition, that

► **Page 10** line −14 — 01 Apr 2017

hear ↝ year (P. Tsanakas)

► **Page 11** line −2 — 01 Apr 2017

$f(n) = e^x$ ↝ $f(n) = e^n$ (P. Tsanakas)

Page 13 line −11 — 12 Aug 2017

big-Oh ↝ big O

► **Page 13** line −8 — 12 Aug 2017

This is called "big-Omega," or $\Omega(n)$, and the precise definition ↝ This is called "big Omega," $\Omega(f(n))$; the precise definition

Page 13 line −6 — 12 Aug 2017

Having defined big-Oh and big-Omega ↝ Having defined big O and big Omega

Page 13 line −5 — 12 Aug 2017

big-Theta ↝ big Theta

► **Page 20** line −4 — 30 Mar 2017

line 3 ↝ line 4

► **Page 20** line −3 — 30 Mar 2017

line 11 ↝ line 12

► **Page 20** line −1 — 30 Mar 2017

line 6 ↝ line 7

Page 40 line 17 — 12 Aug 2017

Using big-Oh notation ↝ Using the big O notation

► **Page 57** line 2 — 24 Apr 2017

When you insert an item in the queue, you increase the index of the head; similarly, when you remove an item from the queue, you increase the index of the tail. ↝ When you insert an item in the queue, you increase the index of the tail; similarly, when you remove an item from the queue, you increase the index of the head. (S. Subramanya)

► **Page 65** line 2 — 06 Mar 2017

011110 ↯→ 011011

► **Page 71** algorithm 3.1, line 1 — 26 Mar 2017

`Size` ↯→ `SizePQ`

► **Page 73** line −11 — 24 Apr 2017

root of the three ↯→ root of the tree (S. Subramanya)

► **Page 80**, line −6 — 25 May 2017

Joyces's ↯→ Joyce's

► **Page 80**, line −5 — 29 Jun 2017

41% ↯→ 53%

Page 91, line −17 — 14 Dec 2017

"1110" ↯→ “1110”

► **Page 95** figure 4.1, caption — 21 Apr 2017

encryption ↯→ decryption

► **Page 140**, line −2 to −1 — 17 Jul 2017

SHA-2 (Secure Hash Standard-2) ↯→ SHA-2 (Secure Hash Algorithm 2)

Page 144, line 2 — 21 Apr 2017

command packet ↯→ *command packet*

► **Page 145**, line −14 — 01 Jun 2017

OR_3 ↯→ OR_2

► **Page 145**, line −12 — 01 Jun 2017

Alice ↯→ OR_1.

► **Page 147**, line −13 — 17 Jul 2017

SHA-224. ↯→ SHA-224,

► **Page 157** figure 6.6, caption — 21 Mar 2017

weigthed ↯→ weighted

► **Page 166** figure 6.13, second panel, label under t — 21 Apr 2017

13 ↯→ $13/-\infty$

► **Page 166** figure 6.13, fourth panel, label under t — 21 Apr 2017

13 ↯→ $13/-\infty$

► **Page 166** figure 6.13, fifth panel, label under t — 21 Apr 2017

$-infty \leadsto -\infty$

Page 178, algorithm 7.1, line 12 — 23 Apr 2017

`ExtractMinFromPQ(`pq`)` $\leadsto$ `ExtractMinFromPQ(`pq`)`

► **Page 179**, line 10 — 24 Apr 2017

line 11 $\leadsto$ line 14 (S. Subramanya)

► **Page 179**, line 12 — 24 Jul 2017

line 11 $\leadsto$ line 14

► **Page 180**, line 13 — 26 Mar 2017

lines 1–7 $\leadsto$ lines 1–10

Page 181, line −4 — 23 Jul 2017

re-weighting $\leadsto$ reweighting

► **Page 182**, figure 7.11 — 22 Jul 2017

link $0 \xrightarrow{0} 2 \leadsto 0 \xrightarrow{4} 2$ and link $0 \xrightarrow{8} 3 \leadsto 0 \xrightarrow{7} 3$

Page 182, figure 7.11, caption — 23 Jul 2017

re-weighted $\leadsto$ reweighted

Page 206, line 1 — 23 Apr 2017

Euros $\leadsto$ euros

► **Page 214**, line 8 — 04 Apr 2017

$P_{B_j} \leadsto B_{P_j}$

► **Page 217**, line −3 — 04 Apr 2017

page 3 $\leadsto$ page 6

► **Page 217**, line −2 — 04 Apr 2017

page 4 $\leadsto$ page 5

Page 222, figure 9.6 — 28 Apr 2017

change line arrow to stealth shape

► **Page 229**, line −16 — 04 May 2017

support $\leadsto$ supported

► **Page 230**, line −3 — 23 Apr 2017

If there are n voters, then candidate A gets $(60 \times 2)n = 120n$ points $\leadsto$ If there are $100m$ voters, candidate A gets $(60 \times 2)m = 120m$ points

► **Page 230**, line −2 — 23 Apr 2017

$(60 + 2 \times 40)n = 140n \leadsto (60 + 2 \times 40)m = 140m$

► **Page 230**, line −2 — 23 Apr 2017

$40n \leadsto 40m$

► **Page 231**, heading 10.2 — 23 Apr 2017

Shulze ⇝ Schulze

► **Page 233**, algorithm 10.1, line 4 — 23 Apr 2017

$P[i][j] \leadsto P[i, j]$

► **Page 234**, line −8 — 04 May 2017

$P[i, j] \leadsto P[c_i, c_j]$

► **Page 234**, line −7 — 04 May 2017

$P[j, i] \leadsto P[c_j, c_i]$

► **Page 234**, line −6 — 04 May 2017

$P[i, j] - P[j, i] \leadsto P[c_i, c_j] - P[c_j, c_i]$

Page 236, line −4 — 28 Apr 2017

$(k + 1) \leadsto k + 1$

► **Page 238**, algorithm 10.2, line 6 — 23 Apr 2017

$S[i][j] \leadsto S[i, j]$

► **Page 238**, algorithm 10.2, line 9 — 23 Apr 2017

$S[i][j] \leadsto S[i, j]$

► **Page 241**, algorithm 10.3, second line of output — 23 Apr 2017

$s[i, j_k] > s[j_k, i] \leadsto S[i, j_k] > S[j_k, i]$

Page 244, algorithm 10.4 — 23 Apr 2017

all $pred$ and $dist$ ⇝ *pred* and *dist*

► **Page 249**, algorithm 11.1 — 24 Apr 2017

a array of items ⇝ an array of items (S. Subramanya)

► **Page 249**, algorithm 11.1 — 24 Apr 2017

a element we are searching for ⇝ an element we are searching for (S. Subramanya)

Page 249, figure 11.1 — 28 Apr 2017

Change the array to [114 , 480 , 149 , 903 , 777 , 65 , 680 , 437 , 4 , 181 , 613 , 551 , 10 , 31 , 782 , 507]; we need not use sequential search in a sorted array.

► **Page 254**, line −5 — 24 Apr 2017

figure 11.3 ↝ figure 11.6

► **Page 260**, algorithm 11.2 — 24 Apr 2017

a element we are searching for ↝ an element we are searching for (S. Subramanya)

► **Page 260**, algorithm 11.2, line 10 — 24 Apr 2017

NULL; ↝ NULL

► **Page 261**, algorithm 11.3 — 28 Jul 2017

`TranspositionSearch(`A, s`)` ↝ `TranspositionSearch(`L, s`)`

Page 261, algorithm 11.3 — 24 Apr 2017

a list of items, ↝ a list of items

► **Page 261**, algorithm 11.3 — 24 Apr 2017

a element we are searching for ↝ an element we are searching for (S. Subramanya)

► **Page 261**, algorithm 11.3, line 12 — 25 Apr 2017

NULL; ↝ NULL

► **Page 262**, algorithm 11.4 — 24 Apr 2017

a array of items ↝ an array of items (S. Subramanya)

► **Page 262**, algorithm 11.4 — 24 Apr 2017

a element we are searching for ↝ an element we are searching for (S. Subramanya)

► **Page 264**, algorirthm 11.5 — 25 Apr 2017

`SecretarySearch(`A, s`)` ↝ `SecretarySearch(`A`)`

► **Page 264**, algorithm 11.5 — 24 Apr 2017

a array of items ↝ an array of items (S. Subramanya)

► **Page 264**, algorirthm 11.5, line 4 — 24 Apr 2017

`Compare(`$A[i], A[b]$`)` ↝ `Compare(`$A[i], A[c]$`)` (S. Subramanya)

► **Page 264**, algorirthm 11.5, line 6 — 25 Apr 2017

$i \leftarrow m + 1$ ↯ $i \leftarrow m$

► **Page 267**, line 18 — 6 May 2017

Unless you are not psychic ↯ Unless you are psychic

► **Page 268**, algorithm 11.6 — 24 Apr 2017

a element we are searching for ↯ an element we are searching for (S. Subramanya)

► **Page 270**, figure 11.14b, last row — 31 May 2017

$l = 7$ ↯ $l = 8$
$m = 7$ ↯ $m = 8$ (I. Kafetzaki)

► **Page 276**, line −2 — 02 May 2017

one's complement ↯ ones' complement

► **Page 278**, algorithm 11.7 — 24 Apr 2017

a element we are searching for ↯ an element we are searching for (S. Subramanya)

► **Page 287**, algorithm 12.1 — 24 Apr 2017

a array of items ↯ an array of items (S. Subramanya)

► **Page 289**, algorithm 12.2 — 24 Apr 2017

a array of items ↯ an array of items (S. Subramanya)

► **Page 291**, algorithm 12.3 — 24 Apr 2017

a array of items ↯ an array of items (S. Subramanya)

► **Page 298**, caption of figure 12.6b — 28 Apr 2017

1 ↯ one

► **Page 299**, algorithm 12.4 — 24 Apr 2017

a array of items ↯ an array of items (S. Subramanya)

► **Page 310**, figure 12.12, third panel — 08 May 2017

$i \rightarrow 5$ ↯ $i \rightarrow 37$

Page 333, line −11 — 09 May 2017

minimal perfect mapping ↯ *minimal perfect mapping*

Page 340, line −3 — 09 May 2017

456, 976 ↯ 456,976

Page 343, figure 13.5 — 09 May 2017

4, 847 ↝ 4,847

Page 343, figure 13.5 — 09 May 2017

126, 033 ↝ 126,033

Page 343, figure 13.5 — 09 May 2017

3, 276, 872 ↝ 3,276,872

Page 346, line 3 — 09 May 2017

binary fractional number ↝ *binary fractional number*

► **Page 353**, line −12 — 23 Jul 2017

An successful search ↝ An unsuccessful search

Page 359, line −9 — 13 May 2017

z-values ↝ z-values

Page 359, line −9 — 13 May 2017

z-axis ↝ z-axis

Page 361, line 7 — 31 May 2017

the number of frequency peaks in the song, and there is even a notation for it: ↝ being the number of frequency peaks in the song, and there is even a notation for it:

Page 361, line 16 — 31 May 2017

move “of” to the next line

► **Page 362**, line −1 — 31 May 2017

the data are not the ↝ the data are not in the

Page 367, line 7 — 13 May 2017

$(1-1/m)^{m(\frac{k}{m})}$ ↝ $(1-1/m)^{m\left(\frac{k}{m}\right)}$

► **Page 370**, figure 13.20, third panel — 13 May 2017

The solid arrows should emanate from “this”.

Page 383, table 14.1 — 14 May 2017

letter ↝ letters

Page 385, line 3 — 14 May 2017

Move J. to next line.

► **Page 386**, line 9, 12, 19 — 25 May 2017

Gibb’s ↝ Gibbs’s

Page 387, line 25 — 16 May 2017

“ineligible” ↝ “ineligible.”

► **Page 390**, line 3 — 16 May 2017

six ↝ five

► **Page 396**, figure 14.8, fourth panel — 17 May 2017

$H = 0.40$ ↝ $H = 0.940$

► **Page 397**, line −9 — 16 May 2017

tox ↝ to

► **Page 400**, figure 14.10 — 08 Jun 2017

{1, 2, …, 14 }: outlook ↝ {1, 2, …, 15}: outlook (V. Malandrakis)

Page 414, line 3 — 12 Aug 2017

because in terms of the big-Oh notation it is ↝ because in terms of the big O notation they are

Page 417, line −3 — 26 Feb 2017

Witten, Frank, and Hall ↝ Witten, Frank, Hall, and Pal

Page 430, line −17 — 23 May 2017

at the start of a string ↝ at the start of the string

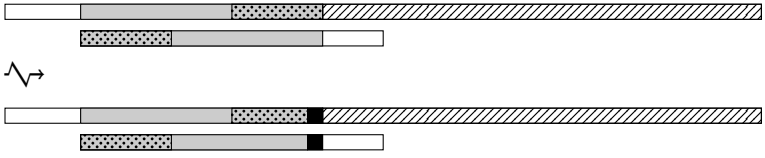
Page 430, line −16 — 23 May 2017

at the end of a string is its *suffix* ↝ at the end of the string is a *suffix*

► **Page 430**, line −4 — 14 Sep 2017

all A, AB, and ABA are ↝ substrings A and ABA are (P. Mpellos)

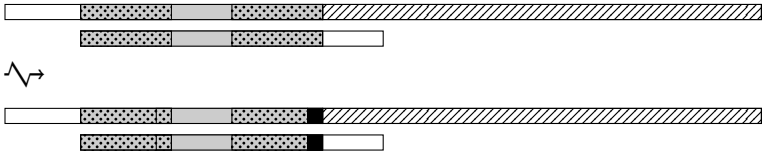
► **Page 431**, fourth graphic — 23 May 2017



► **Page 431**, line −10 — 23 May 2017

of the pattern ↝ of the matched pattern

► **Page 431**, fifth graphic — 23 May 2017



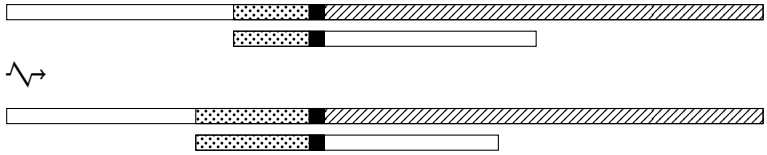
Page 431, line −1 — 24 May 2017

longer shifts ↝ *longer* shifts

► **Page 432**, line −9 — 24 May 2017

So we get: ↝ So we get, indicating the mismatched character:

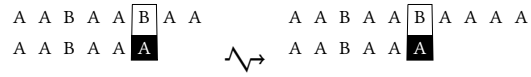
► **Page 432**, second graphic — 23 May 2017



► **Page 432**, line 7 — 24 May 2017

AABAABAA ↝ AABAABAAAA

► **Page 432**, third graphic — 24 May 2017



► **Page 432**, line −4 — 24 May 2017

define its length to be zero ↝ define its border length as zero

► **Page 433**, line 13 — 25 May 2017

borders array ↝ border array

► **Page 434**, algorithm 15.2, line 9 — 02 Jun 2017

$p[i]$ ↝ $p[j]$ (A. Tsalapatis)

► **Page 435**, figure 15.5 caption — 24 May 2017

Another trace the Knuth-Morris-Pratt algorithm; the borders array is at the bottom. ↝ Another trace of the Knuth-Morris-Pratt algorithm; the border array is at the bottom.

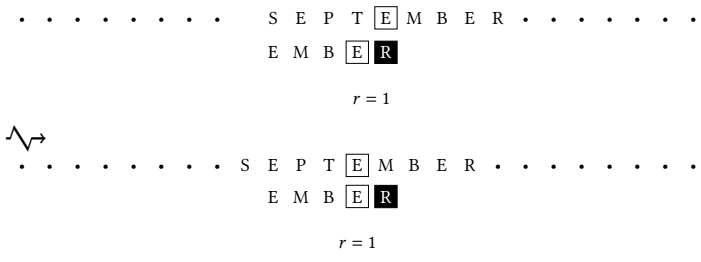
► **Page 437**, line 3 — 25 May 2017

borders array ↝ border array

► **Page 440**, line 12 — 30 May 2017

mattern ↝ pattern

► **Page 441**, figure 15.3b — 30 May 2017



Page 449, line 16 — 23 May 2017

50-50 ↯ 50–50

► **Page 462**, line 10 — 20 May 2017

line 6 ↯ line 7

► **Page 463**, line 4 — 20 May 2017

change ↯ maybe fix

► **Page 466**, lines 18, 21, 23 — 20 May 2017

ECC ↯ EEC

► **Page 467**, lines 12, 19, 23 — 20 May 2017

ECC ↯ EEC

► **Page 467**, paragraph −2 — 22 May 2017

Rewrite the paragraph as follows:

To tackle this kind of question, we must adopt a systematic approach. We have a set of voters, $V = \{v_1, v_2, \ldots, v_n\}$, and a set of weights, $W = \{w_1, w_2, \ldots, w_m\}$. A voter v_i has a weight w_j given by a mapping $f\colon V \to W$. For a decision to be taken, it needs to meet a *quota* Q. In the example of the EEC, we have $Q = 12$. The setup of V, W, f, and Q is called a *voting game*.

► **Page 468**, line 3 — 21 May 2017

such as ↯ such that

► **Page 468**, line 4 — 21 May 2017

in obtaining losing coalition ↯ in obtaining a losing coalition

► **Page 468**, line 14 — 21 May 2017

ECC ↯ EEC

► **Page 468**, line −7 — 21 May 2017

then then ↯ then the

► **Page 468**, lines −3 to −1 — 30 May 2017

As an example, take four voters $V = \{A, B, C, D\}$ with corresponding weights $W = \{4, 2, 1, 3\}$ and quota $Q = 6$. The critical coalitions are (we underline the critical voters) $\{\underline{A}, \underline{B}\}$, $\{\underline{A}, \underline{D}\}$, $\{\underline{A}, \underline{B}, C\}$, $\{\underline{A}, B, D\}$, $\{\underline{A}, C, \underline{D}\}$, $\{\underline{B}, \underline{C}, \underline{D}\}$.

↯

As an example, let us take four voters A, B, C, D with corresponding weights equal to 4, 2, 1, 3, and quota $Q = 6$. The critical coalitions then are, underlining the critical voters: $\{\underline{A}, \underline{B}\}$, $\{\underline{A}, \underline{D}\}$, $\{\underline{A}, \underline{B}, C\}$, $\{\underline{A}, B, D\}$, $\{\underline{A}, C, \underline{D}\}$, and $\{\underline{B}, \underline{C}, \underline{D}\}$.

► **Page 472**, line −1 — 05 Sep 2017

zero ↯ one (N. Batsal)

► **Page 473**, line 1 — 05 Sep 2017

one ↯ zero (N. Batsal)

► **Page 479**, line −4 — 21 May 2017

primes ↯ composites

► **Page 479**, lines −4 to −3 — 21 May 2017

$n(1/2 + 1/3 + 1/5 \cdots + 1/k)$ ↯ $n(1/2 + 1/3 + 1/5 + \cdots + 1/k)$

► **Page 479**, line −3 — 21 May 2017

$(1/2 + 1/3 + 1/5 \cdots + 1/k)$ ↯ $(1/2 + 1/3 + 1/5 + \cdots + 1/k)$

Page 485, output — 23 May 2017

Output: (r, q), such that $n = 2^r q$ ↯ **Output**: (r, q), such that $n = 2^r q$ with q odd

Page 498, reference 219 — 26 Mar 2017

Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 3rd edition, 2011.
↯
Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, Cambridge, MA, 4th edition, 2016.

► **Page 502**, first column — 12 2017

big-Oh ($O(f(n))$ ↯ big O ($O(f(n))$)
big-Omega ($\Omega(f(n))$ ↯ big Omega ($\Omega(f(n))$)
add big Theta ($\Theta(f(n))$), 13

► **Page 503**, second column — 20 May 2017

European Economic Community (ECC) ↯ European Economic Community (EEC)

Page 504, first column, line −15 — 23 Jul 2017

re-weighting ↯ reweighting