Errata for

*Algorithms Sequential & Parallel, A Unified Approach* (Second Edition)
Russ Miller and Laurence Boxer
Charles River Media, 2005

**Chapter 1**

- P. 23, l. 14- to 13-
  
  \( k \) in the worst case, and \( k/2 \) in the average case

  should be

  \( k - 1 \) in the worst case, and \( (k - 1)/2 \) in the average case

- P. 30, l. 5:

  The \( O \)-notation was apparently

  should be

  The \( o \)-notation was apparently

- P. 33, top of page: In the algorithm for MinimumIndex, there are three occurrences of “at” that aren’t, but should be, italicized.

**Chapter 2**

- P. 36, l. 9-10: The list items should be numbered 1), 2), rather than a), b).

- P. 38, l. 2: There should be a period at the end of the line.

- P. 51, 2 paragraphs above Subprogram Split:

  Therefore, the running time of this simple merge algorithm is \( \Theta(k) \), where

  \( k \) is the length of the first input list to be exhausted.

  should be

  Therefore, the running time of this simple merge algorithm is \( \Theta(k) \), where

  \( k \) is the number of nodes (from both input lists) that have been merged
  when the first input list is exhausted.

- P. 56, l. 6-: In the function header, the argument \( n \) should be italicized.
Chapter 3

- P. 61, l. 3: “Let $f(n)$, be” should be “Let $f(n)$ be”

- P. 64, l. 6- - 5- (colon for period):

\[ ... \text{depends on the second summation.} \]

\[ g(n) = \Theta \left[ \sum_{\substack{0 \leq k \leq \log_b n - 1, \\ n / b^k \geq N}} a^k f \left( \frac{n}{b^k} \right) \right]. \]

should be

\[ ... \text{depends on the second summation:} \]

\[ g(n) = \Theta \left[ \sum_{\substack{0 \leq k \leq \log_b n - 1, \\ n / b^k > N}} a^k f \left( \frac{n}{b^k} \right) \right]. \]
Chapter 4

- P. 84, 2nd paragraph:

  exactly $\log^2 2n$ stages of merging

  should be

  exactly $\log_2 2n$ stages of merging
Chapter 5

- P. 99, ER PRAM Algorithm for Broadcasting:
  
  If \( j + 2^{i-1} \leq n \) then \( P_j \) writes \( d \) to \( P_{j+2^{i-1}} \)

  should be

  If \( j + 2^{i-1} \leq n \) then \( P_j \) writes \( d \) to \( d_{j+2^{i-1}} \)

- P. 100, RAM Minimum Algorithm: italicize “\( x_i \)” in “If \( x_i < \text{min\_so\_far} \)”

- P. 101, Figure 5.4: At Time Step 3, we should have \( T[1] = 4 \), not 15.

- P. 105, the algorithm:

  **Output:** succeeds, a flag indicating whether or not the search succeeds and location

  should be

  **Output:** \textit{succeeds}, a flag indicating whether or not the search succeeds, and \textit{location}

- P. 127 (italics):

  \[ 2n^{1/2}(2n^{1/2} - 1) - n = 3n - 2n^{1/2} \]

  should be

  \[ 2n^{1/2}(2n^{1/2} - 1) - n = 3n - 2n^{1/2} \]

- P. 136, 2\textsuperscript{nd} and 3\textsuperscript{rd} lines after caption:

  \((\log_2 n - i + 1)\) dimensional

  should be

  \((\log_2 n - i + 1)\) -dimensional

- P. 140, Cost/Work paragraph:

  Let \( T_{\text{par}}(n) \) represent

  should be

  Let \( T_{\text{par}}(n) \) represent
Chapter 6
No errata reported.

Chapter 7
• P. 174, l. 3 up: “subcube_prefix” should be italicized.

Chapter 8
No errata reported.
P. 208: Item 7’s “Else If” structure is more easily understood using the following alignment.

Else If \( k \leq |smallList| + |equalList| \) then return \( AM \)
Else {find result in \( bigList \)}
    CreateArray(\( bigList, bigList\_array \))
    return Selection\( k - |smallList| - |equalList|, bigList\_array, 1, |bigList| \)
End Else {find result in \( bigList \)}

P. 209, bullet item discussing Step 4:

We can simplify notation by saying that this step requires less than \( T(n/5) \) time.

should be

We can simplify notation by saying that this step requires \( T(n/5) \) time.

P. 210, item c), 2\textsuperscript{nd} sentence:

Thus, the recursive call to \( Selection(k, smallList\_array, 1, |smallList|, smallList\_array, 1 | \) requires at most \( T(7n/10) \) time.

should be

Thus, the recursive call to \( Selection(k, smallList\_array, 1, |smallList|) \) requires at most \( T(7n/10) \) time.

P. 210, l. 2 up – p. 211, l. 3: Delete the two sentences

An upper bound on the right side … we have \( T(n) = O(n) \).
Chapter 10

- P. 265, middle paragraph:
  It is easy to see how such an approach yields a $\Theta\left(n^2\right)$ time RAM algorithm for the intersection query problem, …
  should be
  It is easy to see how such an approach yields an $O\left(n^2\right)$ time RAM algorithm for the intersection query problem, …

- P. 269, Item 5:
  $$(a_i, b_j, i, j) \circ (a_k, b_m, k, m) = \begin{cases} 
  (a_i, b_m, i, m) & \text{if } a_i \leq a_k \leq b_i < b_m \\
  (a_j, b_j, i, j) & \text{otherwise.}
\end{cases}$$
  Thus, $A \circ B$ represents $[a_i, b_j] \cup [a_k, b_m]$, provided these arcs intersect, $b \not\in [a_i, b_j]$, and $[a_k, b_m]$ extends $[a_i, b_j]$ to the right more than does $[a_j, b_j]$; …
  Because the intervals are ordered by their right endpoints, …
  should be
  $$(a_i, b_j, i, j) \circ (a_k, b_m, k, m) = \begin{cases} 
  (a_i, b_m, i, m) & \text{if } a_i \leq a_k \leq b_i < b_m; \\
  (a_j, b_j, i, j) & \text{otherwise.}
\end{cases}$$
  Thus, $A \circ B$ represents $[a_i, b_j] \cup [a_k, b_m]$, provided these arcs intersect and $[a_k, b_m]$ extends $[a_i, b_j]$ to the right more than does $[a_j, b_j]$; …
  Because the intervals are ordered by their left endpoints, …

- P. 275: The last paragraph should not be labeled as item d), as it is a part of item c).
Chapter 11

- P. 292, Figure 11.7: There should be an arrow from the words “column \( k / 2 \)” to the horizontal center of the figure:

- P. 292, paragraph following Figure: There’s a bad line break in the equation

\[
S_{k+1}(i, j) = \min \{ S_k(i, j), S_k(i, k+1) + S_k(k+1, j) \}
\]
Chapter 12

- P. 304, last paragraph:
  
  A path … such that \((v_j, v_{i+1}) \in E\) …

  should be

  A path … such that \((v_j, v_{i+1}) \in E\) …

- P. 323, item 1:

  Entry \(A_k(i, j)\) … time \(3k + |k - i| + k - j| - 2\).

  should be

  Entry \(A_k(i, j)\) … time \(3k + |k - i| + |k - j| - 2\).

- P. 324, caption of Fig. 12.22:

  “At time \(t = 1\),” should be “At time \(t + 1\),”

- P. 326: In order to provide the line references that are used in Figure 12.25, the algorithm for the star function should be presented with lines numbered as follows:

  1. Determine the Boolean function \(\text{star}(v_i)\) for all \(v_i \in V\), as follows.
  2. For all vertices \(v_i\), do in parallel
  3. \(\text{star}(v_i) \leftarrow \text{true}\)
  4. If \(\text{root}(v_i) \neq \text{root}\left(\text{root}(v_i)\right)\), then
  5. \(\text{star}(v_i) \leftarrow \text{false}\)
  6. \(\text{star}\left(\text{root}(v_i)\right) \leftarrow \text{false}\)
  7. \(\text{star}\left(\text{root}\left(\text{root}(v_i)\right)\right) \leftarrow \text{false}\)
  8. End If
  9. \(\text{star}(v_i) \leftarrow \text{star}\left(\text{root}(v_i)\right)\)
  10. End For

- P. 326 (sentence following star algorithm):
  
  See Figure 12-25 for an example that shows the necessity of the step marked \{*\}.

  should be

  See Figure 12.25 for an example that shows the necessity of Step 9.

- P. 332, last sentence of first paragraph:

  Therefore, the running time … is \(O(E \log E)\), which is \(O(E \log V)\).

  should be

  Therefore, the running time … is \(\Theta(E \log E)\), which is \(\Theta(E \log V)\).
• P. 332, 2nd paragraph, 2nd sentence:
  Suppose that instead of initially sorting … into decreasing order
  should be
  Suppose that instead of initially sorting … into nondecreasing order

• P. 349, Exercise 9:
  A bipartite graph … with subsets \( V_0, V_1 \)
  should be
  A bipartite graph … with nonempty subsets \( V_0, V_1 \)
Chapter 13

- P. 353, 6 lines from bottom (space):
  \[ nIsPrime \leftarrow true \]
  should be
  \[ nIsPrime \leftarrow true \]

- P. 361, Table 13.1: The column headers are transposed. The left column should have the column header “\( d \)”. The right column should have the column header “\( n_d \)”. 