THE STATE OF THE ART OF COMPUTER PROGRAMMING

by

D. E. Knuth

STAN-CS-76-551
JUNE 1976

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

Approved for public release;
Distribution Unlimited
This report lists all corrections and changes to volumes 1 and 3 of The Art of Computer Programming, as of May 14, 1976. The changes apply to the most recent printings of both volumes (February and March, 1975); if you have an earlier printing there have been many other changes not indicated here. Volume 2 has been completely rewritten and its second edition will be published early in 1977. For a summary of the changes made to volume 2, see SIGSAM Bulletin 9, 4 (November 1975), p. 109--the changes are too numerous to list except in the forthcoming book itself.

On any given day the author likes to feel that the last bug has finally disappeared, yet it appears likely that further amendments will be made as time goes by. Therefore a family of computer programs has been written to maintain a collection of errata, in the form printed here, but encoded as an ad-hoc sequence of ASCII characters. The author wishes to thank Juan Ludlow-Saldivar for the enormous amount of help he provided in order to get this system rolling. (Some readers who have access to the Stanford A.I.-Lab computer may wish to consult the change file before they report a "new" error; the file name is ACP.MAS [ART, DEK]. Entries for page nnn of volume k begin with $\text{\textbackslash kOlnnn}$ (but change the 01 to 30 if nnn is the Arabic equivalent of a Roman numeral); since "b" is the control character "\^C", you may rather search for simply the string "\text{\textbackslash kOlnnn}". The text of the correction usually includes special codes following the symbol "$\mid$", for things like font changes, etc.)

The author thanks all the bounty hunters who have reported difficulties they spotted. The reward to first finder of each error is still $1 for the first edition and $2 for the second, gratefully paid. Volume 4 remains rather far from completion, so there is plenty of time to work all the exercises in volumes 1-3 and to catch all the remaining errors therein.

This research was supported in part by National Science Foundation grant MCS 72-05752 A05 and by the Office of Naval Research contract NO0014-74-C-0537. Reproduction in whole or in part is permitted for any purpose of the United States Government.
The Art of Computer Programming
Errata et Addenda May 14 1976

2.496 line 5

forcing himself \( \rightarrow \) being encouraged

2.496 line 10

answer \( \rightarrow \) answers

2.496 new quote for bottom of page

We can face our problem.
We can arrange such facts as we have
with order and method.
--HERCULE POIROT, in Murder on the Orient Express (1934)

2.14 line 22

E0. \( \rightarrow \) EO. (boldface)

2.240 line 3

prove \( A6 \rightarrow \) prove that \( A6 \)

2.240 line 1

\( 3n_0 \rightarrow 3n \)

2.240 lines -3 and -2

\( T < 3n_0 \), where \( n_0 \) is the original value of \( n \), \( \rightarrow \) \( T \leq n \)
ex 25

delete step 1.5 and move the 1 to the end of step 1.4

ex 25, change step 1.3 to:

L3. [Shift.] If $x-z < 1$, set $z \leftarrow z$ shifted right 1, $k \leftarrow k+1$, and repeat this step.

line 15, new sentence

hardware. ~ hardware. The idea goes back in essence to Henry Briggs, who used it (in decimal rather than binary form) to compute logarithm tables, published in 1624.

line 23

e.g., example, ~ example

exercise 40

a period (.) should appear after the displayed equation

line 2 two changes

(i) the (q/p) and (p/q) don't match each other. (ii) the first two lines of p44 should be moved back to p43, otherwise the reader will think exercise 47 is complete without turning the page.

line 20

$1/12n \sim 1/(12n)$

ex 15

put spaces in the first matrix, i.e.

$abc \sim a \ b \ c$

$d ef \sim d \ e \ f$

$ghi \sim g \ h \ i$
line 7 after Table 1
Shih-chih  $\leadsto$ Shih-Chih

left side of eq. (17)
move the $k$ a little left, to center it

line 7 after (26)
Shih-chih  $\leadsto$ Shih-Chih

line 8 after (26)
the boldface $3$ appears to be in wrong font (too small)

14 places
change $B$ to $B$ (Roman type) in the notation for Beta function, namely in line 1, line 2, line 3 (twice), line 4 (twice), line 5 (twice), line 7, line 10 (twice), line 12, line 15.

exercise 47
in displayed formula: change upper indices from $n, n+1/2$, $2n+1$, $2n+1-k$ to $r$, $r-1/2$, $2r$, $2r-k$ respectively
line 3: $n = -1$.  $\leadsto r = -1/2$.

lines -3 and -2
before the Renaissance.  $\leadsto$ during the Middle Ages.

line 2

between (23) and (24)
series  $\leadsto$ series (cf. (17))
insert new sentence just after (26):


replace (25) by new equation (25):

\[(1/(1-z)^{m+1}) \ln (1/(1-z)) = \sum_{k \geq 0} (H_{m+k} - H_m) (m+k)^{k^2} \ z^k, \ m > 0.\]

move the copy for each step to the left next to the step numbers (standard format, see e.g. Algorithm E on p2)

line -4

\[\sum \rightarrow \Sigma_k\]

lines 3 and 4 after Fig. 11

\[X; \text{that } \sim X \text{— that values, we } \sim \text{values — we}\]

distribution, the \(\sim\) distribution, we can improve significantly on Chebyshev’s inequality: The

line after (13)

\[f^{(2k+1)}(x) \text{ tends } \sim f^{(2k+1)}(x) \text{ and } f^{(2k+3)}(x) \text{ tend}\]

line 11

\[C \sim C \text{ (Roman, not italics)}\]

line 20

\[\text{records } \sim \text{ blocks}\]
line 5 (two places)

record block

row 5 column 4 of the table

I+T \rightarrow I+T

Fig. 14 in both steps P7 and P6

PRIME[K] \rightarrow PRIME[K]

line 4

fix broken type in the [ of PRIME[M]

line 9

delete the exclamation point (!)

ex 3, first line of program

X+1 \rightarrow X+1(0)

last line of ex 18

assume \rightarrow assume that

line 5

insert more space after the period, this line's too narrow

line no. 21 of the program

PERM+1 \ldots \rightarrow \text{PERM+1,}
line 8
itself " " itself."

line 14
the 0 is broken

line 16
O. J. ~ O.-J.

line -10
print) ~ print),

Fig. 3(a)
delete the funny little box which appears between "third from top" and "fourth from top"

just after (1)
remove black speck

lines -3 and -2
delete the sentence "Is there ... obtainable?"

bottom line
TOP ~ TOP (twice)
line 3
\<L< \sim \<L<

after step names G1 and G2

broken type \(\sim\) for [

line -1

\text{BASE, BASE+1, BASE+2, } \sim \text{ BASE+1, BASE+2, BASE+3,}\

in (10)

move the heavy bar to the right so that it is aligned vertically with the heavy bar in (11)

comment for line 18 of the program

\(T_3 \sim T_4\)

new paragraph before the exercises

In spite of the fact that Algorithm T is so efficient, we will see an even better algorithm for topological sorting in Section 7.4.

changes to Program A

line 04: \(5H \sim 1H\)

line 05: becomes line 06

line 06: becomes line 07

line 07: becomes line 05, and delete the "1H" and change \(x \sim 1+m\)

line 12: becomes the following two lines

\begin{align*}
12 & \text{LD} 2 & 1:3 & \text{LINK} & q' & \text{Q-LINK}(Q1) . \\
13 & \text{JMP} & 20 & q' & \text{Repeat.}
\end{align*}

lines 13-35 become lines 14-36

change GB \(\sim 1B\) in what was line 17 (now line 18)
line -1
\( b^3 \leftarrow b^{3-1} \).

line -4
execute \( b \leftarrow \text{execute } b-1 \).

line 12
29 \leftarrow 27 \text{ (twice)}

Table 1, left column
The line for time 0200 is out of place; it belongs just before the line for time 0256.

Fig. 12
The shading in this figure mysteriously disappeared from the 3rd column of nodes, in the second edition. (First edition was OK.)

line 7
2,192,000 \leftarrow 2,119,200

two lines before (11)
is the lowest value \leftarrow points to the bottom-most value

exercise 20 line 3
\( A(I,J) \leftarrow A[I,J] \)
new exercise

21. [20] Suggest a storage allocation function for $n \times n$ matrices where $n$ is variable. The elements $A[1, J]$ for $1 < I, J < n$ should occupy $n^2$ consecutive locations, regardless of the value of $n$.

tree illustration near bottom of page

the number "9" must be inserted at the right of this diagram

line -13

$P^*$ $\rightarrow$ $P^*$

between (2) and (3)

tilt the diagram $\rightarrow$ and we have $\rightarrow$

tilt the diagram and bend it slightly, obtaining

Fig. /7/

the photograph has been assembled the two parts of this figure improperly in this edition; the left-hand half of the illustration should be lowered so that the trees are flush at the bottom -- this means that corresponding letters will be on the same line in both left and right parts of the illustration

line -9

of (7) $\rightarrow$ of the left-hand tree in (7)

line 16

node to $\rightarrow$ node with

line 9

only upward links are sufficient $\rightarrow$ upward links are sufficient by themselves
1.357  in (17)  

delete "." outside the boxes (for consistency in style)

1.360  exercise 11  

change script \( I \) to italic \( i \) in five places (lines 5, 6, 6, 23, 35)

1.369  Theorem A part (a)  

\( i \sim \).

1.375  line 4  

remove hairline between "fin" and "(".

1.385  line -3  

or it \( \sim \) or

1.385  line -2  

Exercise \( \sim \) exercise

1.405  exercise 12  

Suppose \( \sim [20] \) Suppose

1.406  line -14  

particular \( \sim \) particular

1.427  line -4  

\( 31 \sim 3 ) \)
the shape of the box containing B6 should have rounded sides (like that of B2); on the other hand, the box that says "Error" should be rectangular.

this displayed line should be raised half a space so that it is separated from line -4 by the same amount as it is separated from line -6.

audition \to condition
emergencies \to emergencies.
hence \to Hence

two level \to two-level

exercise 39 line 3

N(n,m) \to N(u,m)/n

line 18

2 \to 2.2

first line of quote

me that \to me ... that
exercise 3

line 3: let \( r \) be \( \rightarrow \) let \( m \) be
line 4: if \( r \neq 0 \), \( \rightarrow \) if \( m \neq 0 \),
line 5: \( \frac{n}{r} \rightarrow n/m \)
\( r \) and let \( m \) be \( \rightarrow \) \( m \) and let \( n \) be
lines 6 and 7 (steps F4 and F5) deleted
line 8: F6. \( \rightarrow \) F4.

better answer to exercise 3

3. -1/27, but the text hasn't defined it.

exercise 13

first sentence should become:
Add "\( T < 3(n-d)+k \)" to assertions A3, A4, A5, A6, where \( k \) takes the respective values 2,3,3,1.

line 16

elements \( a \) and \( b \) \( \rightarrow \) elements, \( a < b \),

exercise 3

the value 3 is \( \ldots \) two \( n^2 \). \( \rightarrow \) \( n^2 \cdot 3 \) occurs for no \( n \), and in the second place \( n^2 \cdot 4 \)
occurs for two \( n \).

line 10

388. \( \rightarrow \) 388; V. S. Linsky, Zh. Vych. Mat. i Mat. Fiz. 2 (1957), 90-119.

new answer replacing answer 10

9.10. No, the applications of rule (d) assume that \( n \geq 0 \). (The result is correct for \( n = -1 \)
but the derivation isn't.)
1/4 \to 1/8 (twice)

exercise 31

We have \[ [This \text{ sum } \text{ was first obtained in closed form by J. F. Pfaff, Nova \textit{acta} \textit{acad. scient. Petr.} 11 (1797), 38-57.] \] We have

through page 487

change \( B \) to \( \text{B} \) (Roman type) in the solutions to exercises 40, 41 (twice), 42, 48 (twice).

exercise 14

\[ n+4 \to n+1 \]

exercise 15

line 1: \( zC_{n-2}(z) \to zC_{n-2}(z)+n0 \)

line 3 (the displayed formula: delete the period, then add a new line: when \( z \neq -1/4; G_{n}(-1/4) = (n+1)/2^n \) for \( n > 0 \).

bottom of page, a new answer to exercise 1.2.1.2-3:

3. \( R_{2k} < |n2k!/|2k|!| \int_1^n \sqrt{(2k)|} dx \). [C. H. Reusch observes that \( R_{2k} = \int_1^n (H_{2k+2} - H_{2k+2}(x)) f^{(2k)}(x) dx/(2k+2)! \), and that \( H_{2k+2} - H_{2k+2}(x) \) always lies between 0 and \( (2-2^{-2k+1})/(2k+2) \). Therefore if \( f^{(2k+1)}(x) \) but not \( f^{(2k+3)}(x) \) tends monotonically to zero, (13) still holds for some \( \theta \) with \( 0 < \theta < 2 - 2^{-2k+1} \).]

exercise 6

\[ O(n^{-3}) \to O(n^{-3}) \]
exercise 14

line 3: MOVE  ~  MOVE
line 4: JSJ*+1  ~  JSJ*+1

exercise 17(b)

(Using assembly ... section.)

(A slightly faster, but quite preposterous, program uses 993 STZ's: JMP 3395; STZ 1,2; STZ 2,2; ...; STZ 993,2; J2N 3399; DEC2 993; J2NN 3001; ENN1 0,2; JMP 3000,1.)

exercise 18

(Unless the program itself appears in locations 0000-0015.)

exercise 20

Fukuoka)

exercise 16 line 1

(49) :  ~  (49);

new line just before answer no. 23:

For small byte size, the entries 2013 would not appear.

exercise 6 line 3

√n  ~  √n

line -13

e.g. the  ~  e.g., the
exercise 22(d)

Since the $a$'s are independently chosen, the ✓ The

exercise 23

line 1: $\int_0^1 \cdots \text{ln} t. \sim \int_0^C \exp(-t-E_1(t)) dt$, where $E_1(x) = \int_x^\infty e^{-t} dt/t$.
line 4: $\ln n / e^T \sim e^{-T} \ln n$
line 6: $83100 \sim 83724 -1795 \sim \text{Math. Comp. 25 (1968), 411-415}$

line 6

dev $\sqrt{1/m}$, ✓ dev $\sqrt{1/m}$, when $n > 2m$.

line 5

process would loop indefinitely; ✓ algorithm breaks down (possibly refers to buffer while I/O is in progress);

exercise 9

in reverse, we can get the inverse ✓ backwards, we can get the reverse of the inverse of the reverse

exercise 12

$0 < a < 1 \sim |a| < 1$

line 12

$r_2(z) \sim r_2(z^x)$

exercise 4(ii) should have the following answer instead:

(ii) LDA $X, 7: 7 (0: 2)$. 


new answer

13. D. J. Kleitman has shown that \( \lim_{n \to \infty} 2^{-n} \log f(n) = \lim_{n \to \infty} 2^{-n} \log \prod_{\theta \in \mathbb{C}, \theta(n)} \).

[To appear.]

line -5

\(-\) COUNT

and also page 544, answer to exercise 24

replace lines 85-87 of the MIX program by

\[
\text{STG } X, 1 \text{ (QLINK)}
\]

\[
\text{QLINK}(r1) \leftarrow k.
\]

Then renumber lines 88-118 to 86-116.

Finally delete "Note: When the ... as the loop." on p. 544.

lines 11-12 change to (with same indentation):

T10. If \( P \neq A \), set QLINK[SUC(P)] \(-\) k, \( P \leftarrow \text{NEXT}(P) \), and repeat this step.

exercise 16

line 2: 295 \(-\) 275 (twice)
line 8: 6 \(-\) 4

line -4 insert new sentence (no new paragraph)

[See exercise 5.2.3-29 for a faster algorithm.]

exercise 1 line 4

AVAIL \(-\) Y \(-\) INFO(P), AVAIL.

line 2

\(-\) COL(P)

17
change answer 18 (saving space for new answer 21):

the first part up to "after x..." can be shortened as follows.

18. The three profit stems respective columns 3,1,2, yield respectively

( )

(use the same matrix.)

exercise 20

A(1,1) \rightarrow A(1,1)

new answer

21. For example, M = max(i,j), LOC(A[i,j]) = LOC(A(1,1)) + M-1 + 1 - j.

(Such formulas have been proposed independently by many people. A. L. Rosenberg and H. R. Strong have suggested the following k-dimensional generalization: LOC(A[i,j,...,I])

\* L_k where L_1 = LOC(A(1,...,1)) + 1 - i, L_r = L_r-1 * (M_r-1)^{r-1} - (M_r-1)^{r-1}, where M_r = max(i,...,i). [IBM Tech. Disclosure Bull. 14 (1972), 3026-3028.]

exercise 15

remove brackets in first and second lines

exercise 12 line 2

A[m] \rightarrow A[m].

new answer

13. (Solution by S. Araujo.) Let steps T1 through T4 be unchanged, except that a new variable Q is initialized to A in step T1; Q will point to the last node visited, if any. Step T5 becomes two steps: T5. (Right branch done?) if RLINK(P) = A or RLINK(P) = Q, go on to T6; otherwise set A = P, P = RLINK(P) and return to T2. T6. [Visit P.] "Visit" NODE(P), set Q = P, and return to T4. A similar proof applies.

line 14

LOC (T) \rightarrow LOC (T).
exercise 1 line 1
consist ~ consists

exercise 12 line 2
INFO(P2)-1 ~ TREE(INFO(P2)-1)

exercise 18 line 5
preorder ~ postorder

exercise 7

the diagrams for Case 1 have two arrowheads in the wrong direction ... the arrows should lead away from a^" and towards b^" both Before and After

line 8
332 ~ 322

exercise 12 line 5

a(i) set a(i) = c(i,j) and b(i) = j; ~
a(j) set a(j) = c(i,j) and b(j) = i;

exercise 16

line 2: the existence of ~ tracing out
lines 4,5: we have an oriented subtree ~ the stated digraph is an oriented tree
line 5: configuration ~ digraph
line 6: subtree ~ tree

last line

exercise 24 line 2
C ~ C

last line of exercise 23, add:

[For \( m \neq 2 \) this result is due to C. Faye Sainte-Marie, *L'Intermédiaire des Mathématiciens* 1 (1891), 107-110.]

exercise 3 line 3
upper ~ right

exercise 10
height ~ weight (three times)

second-last line before exercise 6
this line isn't right-justified, add space after the semicolon

bottom line

Note that there's no comma between Steele and Jr. in his name.

lines 19-21 replace by

Several beautiful Last-copying algorithms which make substantially weaker assumptions about Last representation have been devised. See D. W. Clark, *CACM* 19 (1976), to appear, and J. M. Robson, *CACM* 19 (1976), to appear.

line 4
minuscule ~ minuscule
1.643 line before exercise 34

165. \( \rightarrow \) 165. See also E. Wegbreit, Comp. J. 15 (1972), 204-208; D. A. Zave, Inf. Proc. Letters 3 (1975), 167-169.

1.642 line -7

\( \operatorname{det}(A) \) \( \rightarrow \) \( \operatorname{det}(A) \)

1.649 in several places

change \( = \) to \( \cdots \) in the definitions of upper \( k \), lower \( k \), factorial, and Stirling numbers of both kinds

1.650 bottom line

give section reference 1.25 in right-hand column

1.651 definition of Beta function

\( B \) \( \rightarrow \) \( B \)

1.654 line -20 (the entry for 1 degree of arc)

1154 \( \rightarrow \) 1155

1.655 insert new paragraph after line 7:

See the answer to exercise 1.3.3-23 for the 40-digit value of another fundamental constant.

1.657 last line

9 \( \rightarrow \) 9.
Araujo, Saulo, 560.

Bendix G20, 120.

Breggs, Henry, 26.

Bolzano entry
delete "theorem."

Carlyle, Thomas, xvi.

Christie-Mallowan, Dame Agatha Mary Clarissa (Miller), xix.

Chu Shih-Chih, 52, 58.

Clark, Douglas Wells, 594.

Chebyshev's inequality entry
add p. 102
l,620
Dawson, Reed, 578.

2,621
Doyle, Sir Arthur Conan, 463.

2,622
Even, Shimon, 239.

2,623
Flye Sainte-Marie, Camille, 580.

2,624
delete Fisher, David Allen

2,625
Hamlet, Prince of Denmark, 228.

2,626 entry for Good, Irving John
add p. 578

2,627 line -8

Excursen exercise

2,628
Kleitman, Danil J., 511.
Knopp entry
add p. 491

Knopp entry
fix broken type

line 1
20 ~ 20.

Lanski, V. S., 470.

Path length, 399-405.

Pfaff, Johann Friedrich, 485.

Philco S2000, 120.

Pairet, Hercule, 112.

RCA 601, 120.
2.651B
Rosenberg, Arnold Leonard, 556.

1.651B Robson entry
add p. 594

2.652L
Shakespeare, William, 228, 465.

1.652L
delete Shih-ch'ih, Chu entry

1.652B
Steele Jr., Guy Lewis (Queux), 594.

2.652L
Strong, Hovey Raymond, Jr., 556.

1.652B
Tarjan, Robert Endre, 239.

1.652B
Wadler, Philip Lee, 594

1.652B last line
delete "theorem," (saves one line)
197

Wegbreit, Elton Iben, 603.

198

Wice, David Stephen, 434, 595.

199

Zave, Derek Alan, 90, 603.

(namely the endpapers of the book)

delete "Table 1"
also make the change specified for page 136

200

line 4 of the Preface

system → systems

201

line 4

forcing himself → being encouraged

202

line 10

answer → answers

203

raise this illustration about 3/8 inch

204
making the quotation format more consistent

line 5: The Prince \- The Prince

line 10: MASON (The Case ... 1951) \- MASON, in The Case of the Angry Mourner (1951)

new exercises

21. [M25] (G. Knott.) Show that the permutation \(a_1a_2 ... a_n\) is obtainable with a stack, in the sense of exercise 2.2.1-5 or 2.3.1-6, if and only if \(C_j \leq C_{j+1}\) for \(1 \leq j < n\) in the notation of exercise 7.

22. [M28] (C. Meyer.) When \(m\) is relatively prime to \(n\), we know that the sequence \((m \mod n)(2m \mod n) ... ((n-1)m \mod n)\) is a permutation of \(\{1,2,...,n-1\}\). Show that the number of inversions of this permutation can be expressed in terms of Dedekind sums (cf. Section 3.3.3).

line -9

45885 \- 45855

lines 5-8 after (38)

Curiously ... situation to the \- An interesting one-to-one correspondence between such permutations and binary trees, more direct than the roundabout method via Algorithm I that we have used here, has been found by D. Rotem [Inf. Proc. Letters 4 (1975), 58-61]; similarly there is a

insert new sentence after (53):

Actually the \(O\) terms here should have an extra \(\%\) in the exponent, but our manipulations make it clear that this \(\%\) would disappear if we had carried further accuracy.

exercise 28, three changes

the average is \- the average \(l_n\) is

sorting," for some obscure reason, \- sorting,"

2\(\sqrt{n}\) ... 1.97\(\sqrt{n}\) \- 2\(\sqrt{n}\). L. A. Shepp and B. F. Logan have proved that \(\lim \inf_{n \to \infty} l_n/\sqrt{n} > 2\) (to appear).
figure 9 step D3
COUNT \( [K_j] \sim \) COUNT \( [K_j] \)
addition to step B2
(if BOUND = 1, this means go directly to B4.)
line -5
the underline shouldn’t be broken
comments for lines 14 and 15 of the program
BOUND \( \sim \) BOUND
line 9
(December, 1974.) \( \sim \) (1974), 287-289.
line 8
\( \log_2 \sim \) \( \lg \)
exercise 15 line 2
subscripts on superscripts are in wrong font
line 1
items; \( \sim \) items,
line 3
\( r15 \sim \) \( r15 \)
line -18

at least one

last line of Table 2

line -2

wise, oracle dangerous, adversary

lines -13, -12, -9, -8

pronouncements outcomes (four changes)

lines -7 thru -3

oracles adversaries
oracle adversary (five changes)
Constructing lower bounds. Theorem M shows that the "information theoretic" lower bound (2) can be arbitrarily far from the true lower bound; thus the technique used to prove Theorem M gives us another way to discover lower bounds. Such a proof technique is often viewed as the creation of an adversary, a pernicious being who tries to make algorithms run slowly. When an algorithm for merging decides to compare $A_i : B_j$, the adversary determines the fate of the comparison so as to force the algorithm down the more difficult path. If we can invent a suitable adversary, as in the proof of Theorem M, we can ensure that every valid merging algorithm will have to make a rather large number of comparisons. (Some people have used the words 'oracle' or 'demon' instead of 'adversary'; but it is preferable to avoid such terms in this context, since 'oracles' have quite a different connotation in the theory of recursive functions, and 'demons' appear in still a different guise within languages for artificial intelligence.)

We shall make use of constrained adversaries, whose power is limited with regard to the outcomes of certain comparisons. A merging method which is under the influence of a constrained adversary does not know about the constraints, so it must make the necessary comparisons even though their outcomes have been predestined. For example, in our proof of Theorem M we constrained all outcomes by condition (5), yet the merging algorithm was unable to make use of this fact in order to avoid any of the comparisons.

The constraints that we shall use in the following discussion apply to the left and right ends of the files. Left constraints are symbolized by

**lines 7 and 16**

*questions $\wedge$ comparisons*

*be answered $\wedge$ result in*

**lines 9, 10, 18**

*oracle $\wedge$ adversary (four changes)*

**line 12**

*then we define $\wedge$ thus, to be $\wedge$ is*

**line 15**

*our oracle $\wedge$ that our adversary*
the oracle \rightarrow he

lines 2, 11, 16, 20, -9, -6
oracle \rightarrow adversary (six changes)

line 1
ORACLE \rightarrow ADVERSARY

line 4
its \rightarrow his

e exercise 10, line 2
oracle \rightarrow adversary

e exercise 23, line 6
oracle \rightarrow adversary

line 2
oracle is asked \rightarrow adversary is about to decide

line 3
The oracle \rightarrow he.

lines 5, 11, 20, 24, 27, 30
Say \rightarrow Decide (six changes)
"oracle", \( \leftrightarrow \) "adversary" as in Section 5.3.2.

Line 13: finding an oracle \( \leftrightarrow \) constructing an adversary
Line 15: oracle declare \( \leftrightarrow \) adversary cause
Lines 17, 20, 23: oracle \( \leftrightarrow \) adversary

Replace the eight lines preceding Table 1 by:

May be subject to further improvement. The fact that \( V_4(7) = 10 \) shows that (11) is already off by 2 when \( n = 7 \).

A fairly good lower bound for the selection problem has been obtained by David G. Kirkpatrick [Ph.D. thesis, U. of Toronto, 1974], who constructed an adversary which proves that

\[
V_4(n) > n + \Delta - 3 - \sum_{0 \leq j \leq n-2} \log((n+2-j)/(n+2)) \log \log((n+2-j)/(n+2)), \quad n \geq 2 \Delta + 1. \quad (12)
\]

Kirkpatrick has also established the exact behavior when \( \Delta = 3 \) by showing that \( V_3(n) = n + \log((n-1)/2.5) + \log((n-1)/4) \) for all \( n \geq 50 \) (cf. exercise 22).

Line 17: A. Schönhage \( \leftrightarrow \) M. Paterson, N. Pippenger, and A. Schönhage
Line 18: has \( \leftrightarrow \) have
Line -1: (12) \( \leftrightarrow \) (13)

Line -7: (13) \( \leftrightarrow \) (14)
Line -5: \( V_4(n) \leftrightarrow V_3(n) \)

Line -21: a homogeneous \( \leftrightarrow \) an oblivious
Line -2 and -1: a homogeneous \( \leftrightarrow \) an oblivious
any homogeneous \( \leftrightarrow \) any oblivious
a suitable oracle.] ∼ an adversary.)

substitute for exercise 22

22. [24] (David G. Kirkpatrick.) Show that when $4 \cdot 2^k < n-1 < 5 \cdot 2^k$, the upper bound (11) for $V_2'(n)$ can be reduced by 1 as follows: (i) Form four "knockout trees" of size $2^k$. (ii) Find the minimum of the four maxima, and discard all $2^k$ elements of its tree. (iii) Using the known information, build a single knockout tree of size $n-1-2^k$. (iv) Continue as in the proof of (11).

caption

A homogeneous ∼ An oblivious

line 3


upper left corner of Fig. 51

there's a dot missing on the second line of the diagram for n=6

line 3 new sentence

A. C. Yao and F. F. Yao have proved that $\hat{M}(\ell, n) = C(\ell, n) = \frac{\ell}{2^\ell}$ and that $\hat{M}(m, n) > \frac{1}{2} \lg(m+1)$ for $m < n$ [J/ACM, to appear].

line 12

15 is in the wrong bold-face font

line 13

RECORD (Q) ∼ RECORD (Q)
delete "[Hint: ... 45.31.1]" since the proof of that theorem is being changed in the second edition of vol. 2

other P. \( \mapsto \) other P.

line 15

to C5. \( \mapsto \) to C5 if \( m > 0 \).

lines -16 and -15

SORT10 \( \mapsto \) SORT10

SORT01 \( \mapsto \) SORT01

bottom line

\( \log_2 \) \( \mapsto \) \( \log \) (twice)

lines -3

"Soundex" \( \mapsto \) contemporary form of the "Soundex"

line 17: formulated \( \mapsto \) popularized

lines 19-20: inversely \( \mapsto \) Reading \( \mapsto \) approximately proportional to \( 1/n \). [The Psycho-Biology of Language (Boston, Mass.: Houghton Mifflin, 1933); Human Behavior and the Principle of Least Effort (Reading]

extra annotation on line 08 of Program B

\( \text{lrA/2.1} \mapsto \text{lrA/2.1} \) (rX changes too)
only all only if all

between and outside the extreme values of the

(6)

\[ 1 < j < 2 \]

line -10 and also line -18

800 memory. The difference between \[ \log \log N \] and \[ \log N \] is not substantial unless \( N \) is quite large, and typical files aren't sufficiently random either. Interpolation

new paragraph after line 14:

Interpolation search is asymptotically superior to binary search: one step of binary search essentially replaces the amount of search, \( n \), by \( \frac{1}{2}n \), while one step of interpolation search essentially replaces \( n \) by \( \sqrt{n} \) if the keys in the table are randomly distributed. Hence it can be shown that interpolation search takes about \( \log \log N \) steps on the average. (See exercise 22.)

replace lines -5 thru -11 by:

<table>
<thead>
<tr>
<th>01 13 09 31 29 08 08 53 20</th>
<th>01 13 14 31 52 30</th>
<th>01 13 43 10 48</th>
<th>01 13 48 10 30</th>
<th>01 14 01 26 40</th>
</tr>
</thead>
<tbody>
<tr>
<td>49 12 27</td>
<td>49 09 07 12</td>
<td>48 49 41 15</td>
<td>48 16 22 59 25 25 55 33 20</td>
<td>48 36</td>
</tr>
</tbody>
</table>

bottom line

was seems to have been
line 11: 1 → 1.2
line 12: February, → February

the last part is in nearly perfect alphabetic order! →
the alphabetic order in the last part is substantially better.

line 2

replace exercise 22

22. [M43] (A. C. Yao and F. F. Yao.) Show that an appropriate formulation of
interpolation search requires asymptotically lg lg N comparisons, on the average, when
applied to N independent uniform random keys that have been sorted. Furthermore, all
search algorithms on such tables must make asymptotically lg lg N comparisons, or the
average.

line -8

A) → A), since the necessary operations are trivial when ROOT = A.

line 17

Algorithm I. → Algorithm T

lines 7 and 8

clearly constructed n+1 different deletions; →
constructed n+1 different deletions, one for each j;

line 6: A fairly → An even more
line -7: time. → time. In fact, M. Fredman has shown that (ln) units of time suffice, if
the right data structures are used [ACM Symp. Theory of Comp. 7 (1975), 240-244].
in the second edition of vol. 3 I must revise the subsection about the Hu-Tucker algorithm to take account of the new Garasch-Wachs algorithm. Meanwhile I could have improved my treatment of Hu-Tucker by leaving the external nodes out of the priority queues (cf. (23) on p. 144, an unnecessarily cumbersome approach).

replace lines 3-5 by:

that the resulting maximum subtree weight, \( \max(w(0,k-1),w(k,n)) \), is as small as possible. This approach can also be fairly poor, because it may choose a node with very small \( p_k \) to be the root; however, Paul J. Bayer has proved that the resulting tree will always have a weighted path length near the optimum (see exercise 36).

exercise 30

M46 \( \sim \) M41

new version of exercise 36

36. [M40] (Paul J. Bayer.) Generalizing the upper bound of Theorem G, prove that the cost of any optimum binary search tree with nonnegative weights must be at most the total weight \( S \leq \Sigma_{i \in N} p_i + \Sigma_{q_i \in N} q_i \text{ times } H + 2 \), where

\[
H = \Sigma_{1 \leq i \leq n}(q_i/S) \log(S/q_i)
\]

in fact, the top-down procedure which repeatedly chooses roots that minimize the maximum subtree weights will yield a binary search tree satisfying this bound. Show further that the cost of the optimum binary tree search tree is \( 2S \text{ times } H - \log(2H/e) \).

diagrams (2)

put extra little vertical lines above the topmost nodes (\( B \) and \( X \), respectively), for consistency with (1)

line 2

K \( \sim \) K
replace lines -4 and -3 by:

indicate that the average number of comparisons needed to insert the Nth item is approximately \( 1.01 \lg N + 0.1 \) except when \( N \) is small.

bottom line of Table 1

2.8 \( \sim \) 2.78

Eq. (14)

\( p/(1-p) \approx 1.851 \), \( \sim \) \( 1/(1-p) \approx 2.851 \).

line -12

\( k - 1 \) is \( p/(1-p) \), \( \sim \) \( k \) is \( 1/(1-p) \).

line 1: \( \lg N \cdot 0.25 \) \( \sim \) \( 1.01 \lg N + 0.1 \)
line 2: \( 11.17 \lg N \cdot 4.8 \) \( \sim \) \( 11.31 \lg N + 3 \)
line 6: \( 6.5 \lg N \cdot 4.1 \) \( \sim \) \( 6.6 \lg N + 3 \)

Figure 24

below the third node from the left, the 1 has a bar across it, making it look like a 4 by mistake.

line -9

\( R(P) \) \( \sim \) \( \text{RANK}(P) \)

line 16

\( \text{RANK}(R) \) \( \sim \) \( \text{RANK}(R) \). Go to C10.
line -4


trees which arise when we allow the height difference of subtrees to be at most $k$. Such structures may be called $H[k]$ trees (meaning "height-balanced") so that ordinary balanced trees represent the special case $H[1]$. Empirical tests on $H[k]$ trees have been done by F. L. Karlton et al., *CACM* 9 (1966), 23-28.

new exercise

31. (M. L. Fredman.) Invent a representation of linear lists with the property that insertion of a new item between positions $m-1$ and $m$, given $m$, takes $O(\log m)$ units of time.

new paragraph before the exercises

Andrew Yao has proved that the average number of nodes after random insertions with the overflow feature will be $\frac{N}{m} \ln \ln 2 + \frac{O(N/m^2)}{m}$, for large $N$ and $m$, so the storage utilization will be approximately in $\frac{2 \approx 69.3\%}{2}$ in *Information, to appear*.

line 11

long but always a multiple of 5 characters,

tree.

line -4

HOUSE HOUSE (twice)
S.454 line 5

the nodes of the tree \( \sim \) the tree is nonempty and that its nodes

S.454 line 19

follows. \( \sim \) follows:

S.504 exercise 4

there will be a new illustration, with positions numbered from 1 to 49 instead of 1 to 55. The respective entries will be:

\[
\begin{array}{cccc}
\text{---} & (20) & \text{---} & \text{WAS THAT (18) OF} \\
\text{BE THE HIS WHICH WITH THIS} & \text{---} & \text{---} & \text{---} \\
(4) & \text{ON} & \text{HE A OR (19)} \\
(3) & \text{TO HAD} & \text{---} & \text{---} \\
(17) & \text{FOR BY IN FROM AND NOT} \\
(1) & \text{HER ARE IS IT AS AT} \\
(7) & \text{---} & \text{HAVE (3)} & \text{---} \text{YOU ---} \\
\end{array}
\]

line 2: 55 \( \sim \) 19

lines after new illustration: 20,1,14,...,2 within \( \sim \) 20,19,3,14,1,17,1,7,3,20,18,4 within

S.505 line 2

that \( \sim \) that, if \( n > 2 \),

S.504 exercise 39

\( M_{47} \sim M_{43} \)

S.515 line -5 add new sentence after "of M."

\( (A \text{ precise} \text{ formula is worked out in exercise 34.)} \)

S.520 program line 13

empty \( \sim \) nonempty


delete lines 15-18 (m=1 not really needed after all)

line -1

actually substantially

line 13

similar weaker

three lines after (34)

purposes. purposes. In fact, Leo Gubas and Endre Szemerédi have succeeded in proving the difficult theorem that double hashing is asymptotically equivalent to uniform probing, in the limit as $M \to \infty$. [To appear.]

just after (37), insert new sentence:

By convention we also set $f(0,0) = 1$.

new formula for (58)

$$C_N = 1 + \frac{(a-b)^2}{(b^2)} (2 + (a-1)b + (a^2-(a-1)b^2)) \cdot O(1/M).$$

line -2

until Morris's ... 1968, until the late 1960's,

line 1: The only ... among The first published appear ... of the word seems to have been in H. Hellerman's book *Digital Computer System Principles* (New York: McGraw-Hill, 1967), p. 152; the only previous occurrence among line 6: 1968 1967
exercise 5 lines 3 and 4

Line -4

exercise 10

exercise 45

exercise 66

Exercise 66

new exercise

67. (M25) (Andrew Yao.) Prove that all fixed-permutation single-hashing schemes in the sense of exercise 62 satisfy the inequality $C_N \geq \frac{M}{2} + 1/(1-a)$. (Hint: Show that an unsuccessful search takes exactly $k$ probes with probability $p_k < (M-N)/M$.)

lines -10, -8, -6

LONGITUDE \nsim \nLONGITUDE (three places)

in the second edition I will be reviewing Section 6.5 again, deleting the material on post-office trees, paying more attention to Bentley's $k$-$d$ trees, and discussing the search procedure of Burkard and its analysis by Dubois and Trausse (cf. Stanford CS report of Sept. 1975)
line 13, add:

[C/CM 18 (1975), 509-516.]

line 8

3 (to appear) \( \rightarrow \) 4 (1971), 1-10

the numbers in (5) should be respectively:

\[
\begin{align*}
0.07048358; \quad 0.00701159; \quad 0.00067094; \quad 0.00000786; \quad 0.00000728; \quad 0.00000082.
\end{align*}
\]

quotation

Alice’s Adventures in Wonderland \( \rightarrow \) Alice’s Adventures in Wonderland

lines 1-3

So we may \( (p-1)/2. \) \( \rightarrow \)
In general if \( f \) is any divisor of \( p-1 \) and \( d \) any divisor of \( \text{gcd}(f,n), \) we can similarly determine \( (n/d) \mod f \) by looking up the value of \( b(p-1)/f \) in a table of length \( f/d. \) If \( p-1 \) has the prime factors \( q_1 < q_2 < \cdots < q_t \) and if \( q_1 \) is small, we can therefore compute \( n \) rapidly by finding the digits from right to left in its mixed-radix representation, for radices \( q_1, \ldots, q_t. \) (This idea is due to R. L. Silverman.)

exercise 6

the \( 1 \)’s in the exponents are too high (twice)

exercise 13

\( b_{m-1}, \leftarrow b_{m-1}, b_{m+1}, \)

exercise 20

22. \[ Lm_j/n_1 - Lm_i/n_1 - Lm(j-i)/n_1 \times 0 \text{ or } 1, \text{ and } \alpha = 0 \text{ iff } mj \mod n > mi \mod n \]
Hence the number of inversions is \[ \Sigma_{0 \leq j \leq n} \{Lm_j/n_1 - Lm_i/n_1 - Lm(j-i)/n_1\} \times \Sigma_{0 \leq r \leq m} \{Lm/r \_ n\} \times (r - (k-r) - (k-r-1)), \]
which can be transformed to \[ \frac{1}{2} \ln(1)(n-2) - \frac{1}{2} \ln(m,n,0). \] [J. für die rein und angew. Math. 198 (1957), 162-166.]

**Exercise 19**

Delete lines 3-7 of this answer.

Line 8: The answer \( N_j \) (This formula now add a new paragraph:

*Note: A general formula for the number of ways to place the integers \{1,2,...,n\} into an array which is the "difference" of two tableau shapes \((n_1, ..., n_m)\) \((\ell_1, ..., \ell_m)\), where \(0 < \ell_j < n_j\) and \(n = \Sigma n_j - \Sigma \ell_j\) has been found by W. Feit [Proc. Amer. Math. Soc. 4 (1953), 740-744]. This number is \(n! \det(1/(n-j) - (\ell_j-i))\).

**Line 4**

\[ 4.5N^2 + 2.5N - 6. \]

**Addendum to Exercise 15**

It is interesting to note that \(G(w,z) = F(-w,z)/F(-w,\bar{z})\), where \(F(z,q) = \Sigma_{n>0} z^n q^n / \prod_{1 \leq k \leq \infty} (1-q^n)\) is the generating function for partitions \(p_1 + ... + p_n\) into \(n\) parts, where \(p_j \leq \sqrt{j + 2}\) for \(1 < j < n\) and \(p_n > 0\) (cf. exercise 5.1-1a).

**Exercise 31 Line 03**

INPUT-N, 4 \( \rightarrow \) -INPUT-N, 4

**Addition to Answer 2**

[Algorithm 5.2.3S does exactly \( \text{ch}(2) \) exchanges, see exercise 5.2.3-1.]

**Line 12**

(to appear) \( \mapsto 11 \) (1975), 29-35.)
bottom two (clobbered) lines should start respectively thus:

43. As $a \to 0^+$, 
$\Gamma(1)/a \to \Gamma'(1) \ast -\gamma$.

line -4

$\Gamma$ is in wrong font (see line -2 for correct $\Gamma$)

exercise :3

397-104 $\sim$ 263-269

lines -8, -7, -6, -3

oracle $\sim$ adversary

lines 2, 17

oracle $\sim$ adversary

exercise 9

comparisons.) $\sim$ comparisons, yet the procedure is not optimal.)

exercise 14

line 1: found in $\sim$ found in $U_1(n) <$
also add new sentence: (Kirkpatrick's adversary actually proves that (12) is a lower bound for $U_1(n+1) - 1$.)

line 2

oracle $\sim$ adversary
22. In general when $2^r < n < (2^r+1)2^k$ and $t < 2^r < 2t$, this procedure starting with $t+1$ knockout trees of size $2^k$ will yield $L(t) - I/2$ fewer comparisons than (11), since at least this many of the comparisons used to find the minimum in (ii) can be "reused" in (iii).

exercise 36 last line

to appear.} 333-339.

insert new paragraph before line -2:

G. Randel and D. Stevenson have observed that exercises 37 and 38 combine to yield a simple sorting method with $(\log n)/k + O(n)$ comparison cycles on $k$ processors: First sort $k$ subfiles of size $< \lceil \log n/k \rceil$, then merge them in $k$ passes using the "odd-even transposition merge" of order $k$. (To appear.)

exercise 2 line 4

D8 348

new answer


exercise 3 for section 5.5, last line

variables. 348 variables, without transforming the records in any way.

line -6

Strauss 349 Strauss

exercise 7 line 3

80]. 350 80; see also L. Guibas, Acta Informatica 4 (1975), 293-298.]
line 9 (displayed nodes):
\begin{align*}
& r_1 \leadsto r_0 \\
& r_2 \leadsto r_1 \\
& s_1 \leadsto s_0 \\
& s_2 \leadsto s_1 \\
\end{align*}
line 8:
\begin{align*}
& r_1 \leadsto r_0 \\
& s_1 \leadsto s_0 \\
& k > 1 \leadsto k > 0 \\
\end{align*}
lines 6 and 5: the right subtrees of \ldots and the result \leadsto the result

new answer

30. This has been proved by Russell Wesser [to appear].

replace answer to 36 by:


exercise 19

the fourth rectangle in the left-hand figure is too short -- it should be extended so that its bottom line is at the same level as the bottom of the first and third rectangles

answer 20, the line following the tree should become:

It may be difficult to insert a new node at the extreme left of this tree.

answer 30 line 4

left subtree of that \leadsto subtree rooted at that

new answer

29. Partial solution by A. Yao: With \( N > 6 \) keys the lowest level will contain an average of \( \frac{2}{N-1} \) one-key nodes, \( \frac{1}{N+1} \) two-key nodes. The average total number of nodes lies between \( 0.70N \) and \( 0.79N \), for large \( N \). [Acta Informatica, to appear.]
new answer

31. Use a nearly balanced tree, with additional upward links for the leftmost part, plus a stack of postponed balance factor adjustments along this path. (Each insertion does a bounded number of these adjustments.)

exercise 4 line 3

IONIC ~ TRASH

seem ~ sir
insert new sentence on last line: [This remarkable 49-place packing is due to J. Scott Focht, who showed that 48 places do not suffice.]

new answer to exercise 11 (extends to p. 683)

11. No; eliminating a node with only one empty subtree will "forget" one bit in the keys of the nonempty subtree. To delete a node, it should be replaced by one of its terminal descendants, e.g., by searching to the right whenever possible.

exercise 12

line 3: Algorithm 6.2.2D. ~ the algorithm suggested in the previous answer.
last line: \( \frac{1}{k} \sim \frac{1}{f} \)

exercise 34 line 1

\( B_k^2 \sim B_k^2 k^{-1} \)

exercise 34, new answer to part (b)

(b) In the \( 1/\exp(-1) \) part, it suffices to consider values of \( j \) with \( x < 2 \) in \( n \). For \( 1 < x < 2 \)

\( n \) we have

\[
\sum_{k=1}^{n} \exp(-kx) \sim B_k^2 k^{-1} e^{-k} + O(x^{-2} e^{-x}/n).
\]

For \( x > 1 \) we have

\[
\sum_{k=1}^{n} \exp(-kx) \sim B_k^2 k^{-1} e^{-k} + O(x^{-2}/n).
\]

line -9

\( \sim f(n) \sim f(n)^{1/2}/n \),
\( k < 1 \sim k > 1 \)


and $\sim$ with $0/0 = 1$ when $k = N \cdot M - 1$, and

in the second edit on line 12 I will revise several of these answers, using Mike Paterson’s simplified new approach to such analyses.

exercise 39

line 6 (third line of displayed formulas): delete "ji" (on this line only)

line 6 (fourth line of displayed formulas): $(\bar{j}) \sim \Sigma j \bar{j} (\bar{j})$ (two places)

new answer


new answer

67. Let $q_k = p_k^* n_{k+1} \cdots$; then $C_N^* = \Sigma k \to q_k$ and $q_k > \max \{0, 1-(k-1)(M-N)/M\}$.

line 1

$\Sigma k p_k^* p_{r_k}^* \sim \Sigma \bar{k} p_k^* p_{r_k}^*$ minus the probability that a particular record is a “true drop”, namely $(N-q)/\bar{N}$, where $N = \{\bar{k}\}.

line -20 last column

$1154 \sim 1155$
A few interesting constants without common names have arisen in connection with the analysis of sorting and searching algorithms; 10-digit values of these constants appear in the answers to exercises 5.2.3-27, 5.2.4-13, and 6.3-27.

\[ \prod \det(A) \triangleleft \det(A) \]

definition of factorial

\[ 1 \cdot 2 \cdots n \triangleleft 1 \cdot 2 \cdots n \]

definitions of \( x \) lower \( k \) and Stirling numbers of both kinds

Adversaries, 200-201, 209, 211-212, 220.

Aho entry

add p. 468
E.7111
Bandiet, Gerard, 610.

E.7112
Bayer, Paul Joseph, 139, 150.

E.7113
Dedekind sums, 22.

E.7114
Deons, 200.

E.7115
Fett, Walter, 592.

E.7116
Fishburn, John Seot, 680.

E.7117
Fredman entry
add pp. 139, 171

E.7118
Grassell  Grasselli

E.7119
Guibas entry
add pp. 528, 612, 696
\textbf{E.7250}
Hellerman, Herbert, 542.

\textbf{E.7250}
Hoshu, Mamoru, 687.

\textbf{E.7250}
dele the entry for "Homogenous comparisons".

\textbf{E.7250} Hyafil entry

delete p. 215.

\textbf{E.7250} two new entries

Hill(s) trees, 468. Height-balanced trees, 468, see Balanced trees.

\textbf{E.7250} Knockout tournament entry

add pp. 214, 220.

\textbf{E.7250} Linear list representation entry

468. \rightarrow 471.

\textbf{E.7250} two new entries

Karilton, Philip Lewis, 468

\textbf{E.7250}
Logan, Benjamin Franklin, Jr., 591.
Meyer, Curt, 22.

Miyakawa, Masahiro, 687.

Oblivious algorithms, 220-221.

Oracles, 200, see Adversaries.

Parallel computation entry

add p. 640

Paterson, Michael Stewart, 217.

Pippenger, Nicholas John, 217.

line -1

78 ← vi, 78

Rotem, Doron, 64.
Sherp, Lawrence Alan, 594.

Sline -7
223, 223, 405 (exercise 22),

Silver, Roland Lazarus, 576.

Simultaneous comparisons entry
add p. 640

Stevenson, David, 640.

new subentry under Sorting
history of, 382-388, 417-418.

Strauss → Straus

Sugito, Yoshio, 687.

Szemerédi, Endre, 528.
417
Tape searching, 100-101, 105.

418
line 25
Slawomir ~ Slawomir

419
Treesort, see Tree selection sort, Heapsort.

420
Two-dimensional trees, 555, 570.

421
Turski entry
Wladyslaw ~ Wladyslaw

422
Ullman entry
add p. 368

423
new subentry under Trie search
generalized, 565.

424
Wessner, Russell, 674.

425
Yao, Fong, Frances, 232, 422.
Wrench entry 426
add p. 686

Yao, Andrew entry 427
add pp. 232, 422, 479, 549, 678

Yuba, Toshibe, 687.

Zeta function, 612, 666.

just before 2-3 trees entry 430
2D trees, 555, 570.

(namely the endpapers of the book) 431
delete "Table 1"
also change 1 to 10 in box number 35

changes to MIX booklet 432
p30, Fig. 3: Step P3 should say "500 found?"
p34, Fig. 4: third card should say L EQU 500
p43, line 1: 6667 $\rightarrow$ 66667
p43, line 2: 193,331 $\rightarrow$ 133,331
p44, problem 16, line 2: row...diagonal $\rightarrow$ row and column
p44, problem 16, line 8: 10 $\rightarrow$ 9
and change "record" to "block" everywhere in the discussion of MIX I/O operators.
\( \xi, \mathcal{U} \) changes to the book "Surreal Numbers"

p99, line 2: (4) \( \rightsquigarrow \) (3)
p111, lines 4 and 5, interchange the inside of the braces:
\[ \{ x - \frac{x^2}{x^2}, x - \frac{x^2+x^3}{x^2}, \ldots \} \]

p117, problem 18, lines 3 and 4 should be:
\( X_L \) has a greatest element or is null if and on
\( X_R \) has a least element or is null.
This report lists all corrections and changes to volumes 1 and 3 of *The Art of Computer Programming*, as of May 14, 1976. The changes apply to the most recent printings of both volumes (February and March, 1975); if you have an earlier printing there have been many other changes not indicated here. Volume 2 has been completely rewritten and its second edition will be published early in 1977. For a summary of the changes made to volume 2, see *SIGSAM Bulletin* 9, 4 (November 1975), p. 10f -- the changes are too numerous to list except in the forthcoming book itself.