Index

Abelson, Harold, 97
abs, 19
absolute path name, 6
acyclic graph, 216
adjacency list, 217
adjacency matrix, 224
agent, 123
AI, see artificial intelligence
ALGOL, 198
algorithm, 58, 197
alias, 46
ALU, 147
anchor text, 255
AND gate, 141
API, 86
append, 219
applet, 193
application layer, 189
application programming interface, 86
arithmetic and logic unit, 147
array, 182, 214
artificial intelligence, 85, 123, 210, 271
artificial life, 269
ASCII, 194
assoc, 107, 218
association list, 214, 218
atomism, 273
awk, 9
backward chaining, 16
bag, 213
bash, 24
Bayes, Thomas, 135
bc, 179
behaviorism, 284
big-endian, 148
bison, 127
bit, 146
Boole, George, 141
Boolean function, 141
Borning, Alan, 97
bot, 123
Bourne, Steven, 24
branch instruction, 154
breadth-first search, 221
brickOS, 172
Bush, Vannevar, 13
byte, 148, 153
bytecode, 116, 193
C, 65, 82, 159, 172, 199
C++, 94, 110
cadr, 219
car, 218
cat, 10
cd, 228
cdr, 19, 218
cellular automata, 269
central processing unit, see CPU
CGI, 192
chdir, 228
class, 108, 110
classification, 131
client-server model, 186
clock cycle, 170
collaborative filtering, 125
combinatorial optimization, 259
combinatorics, 204
command language, 23
comments, 173
Java and C, 173
machine code, 157
Common Lisp, 18
compiler, 159
composite data type, 213
computational complexity, 206
computational geometry, 233
computational model, 71
cond, 111
conditional probability, 134
configuration space, 233
connected graph, 216
cons, 107
consciousness, 276
constraint programming, 95
dependence, 249
Conway, John, 269
Cook, Stephen, 209
CPU, 153, 163
cycle, 169
cross section, 223
crossover operator, 262
csh, 24
curl, 238
cyborgs, 280
DAG, 216
Darwin, Charles, 257
data mining, 55
data structure, 58, 197, 213
data-link layer, 190
database, 48
attribute, 49
join, 53
query, 51
record, 49
table, 49
debugging, 81
decidability, 206
decision tree, 133
define, 64, 74
degree of freedom, 233
Democritus, 273
Dennett, Daniel, 287
depth-first search, 217
determinism, 281
diagnostic characteristic, 124
digital abstraction, 144, 170
Dijkstra, Edsger, 216
directed graph, 214
directory
home, 6
root, 6
working, 26, 228
distribution, 250
uniform, 250, 265
do, 83
documentation, see comments
dot product, 225
dynamic content, 192
doc, 83
documentation, see comments
dot product, 225
dynamic content, 192
echo, 28
echo, 28
egrep, 45
e-mail, 122
environment variable, 29
escape character, 45
eval, 30
exceptions, 171
exec, 172
exponential scaling, 202
expressiveness, 86
exp, 200
factorial, 63
Fibonacci sequence, 199
field, 110
firewall, 2
fitness function, 257, 263
fixed point, 251
flex, 127
flip-flop, 149
flow of control, 25
for loop, 82
formal parameter, 75
Fortran, 17
free will, 281
frequency, 265
FTP, 187, 237
Index

ftp, 187
full adder, 147
function, 75
    argument, 75
    definition, 75
functional programming, 222
gene, 257
genetic algorithm, 257
genetic operator, 262
GIF, 192
gigahertz, 170
GISP, 258
Gödel, Kurt, 206
golden mean, 200
graph, 214
    algorithm, 217
    directed, 215
    search, 217
    topology, 214, 253
    undirected, 215
grep, 43, 239
hacker, 103
half adder, 146
halting problem, 206
ham, 124
hash table, 218
histogram, 265
    word, 240
Holland, John, 257
hosting web pages, 187
HTML, 7, 192
HTTP, 187, 237
hyperlinks, 192
hypothesis, 258
    space, 258
implementation, 58
INBOX, 125
incompleteness, 206
infinite loop, 160, 239
info, ii
information retrieval, 240
inner product, 225
instance, 110
instantiate, 112
instruction address register, 154
interface, 58, 108
interrupt, 171
intractability, 206
inverse document frequency, 245
inverter, 142
IP, 190
ISO 8601, 110, 194
James, William, 285
Java, 65, 93, 110, 239
Java virtual machine, 116, 193
jobs, 39
Joy, Bill, 24
JPEG, 192
JSR, 165
jump instruction, 154
Kant, Immanuel, 279
Karp, Richard, 209
Kay, Alan, 210, 288
kernel, 23
kill, 168
kilobyte, 153
Knuth, Donald, 119, 285
Korn, David, 24
ksh, 24
Kurzweil, Ray, 274
Lamarck, Jean-Baptiste, 260
lambda, 88, 223
LAN, 191, 215
Laplace, Pierre-Simon, 281
latch, 149
Lego Mindstorms, 156, 172
length, 18
let, 88
Levin, Leonid, 209
Lex, 127
lexer, 126
library, 86, 107
Licklider, J. C. R., 13
linear algebra, 225
linear scaling, 202
link, 214
Lisp, 18
list, 105
list-ref, 105
list-tail, 105
<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>little-endian</td>
<td>148</td>
</tr>
<tr>
<td>ln</td>
<td>229</td>
</tr>
<tr>
<td>loader</td>
<td>157</td>
</tr>
<tr>
<td>local variable</td>
<td>88</td>
</tr>
<tr>
<td>logarithm</td>
<td>202</td>
</tr>
<tr>
<td>logic diagram</td>
<td>142</td>
</tr>
<tr>
<td>logic gate</td>
<td>141</td>
</tr>
<tr>
<td>ls</td>
<td>7, 25, 228</td>
</tr>
<tr>
<td>machine language</td>
<td>152</td>
</tr>
<tr>
<td>main</td>
<td>66, 115, 172</td>
</tr>
<tr>
<td>make</td>
<td>16</td>
</tr>
<tr>
<td>makefile</td>
<td>16</td>
</tr>
<tr>
<td>man, ii</td>
<td></td>
</tr>
<tr>
<td>map</td>
<td>19, 222, 264</td>
</tr>
<tr>
<td>Maple</td>
<td>13</td>
</tr>
<tr>
<td>Mathematica</td>
<td>12</td>
</tr>
<tr>
<td>Matlab</td>
<td>13</td>
</tr>
<tr>
<td>matrix</td>
<td>224, 250</td>
</tr>
<tr>
<td>McCarthy, John</td>
<td>123, 210</td>
</tr>
<tr>
<td>McDermott, Drew</td>
<td>283</td>
</tr>
<tr>
<td>megabyte</td>
<td>153</td>
</tr>
<tr>
<td>megahertz</td>
<td>177</td>
</tr>
<tr>
<td>memoization</td>
<td>90</td>
</tr>
<tr>
<td>Mendel, Gregor</td>
<td>257</td>
</tr>
<tr>
<td>method</td>
<td>110</td>
</tr>
<tr>
<td>Mill, John Stuart</td>
<td>286</td>
</tr>
<tr>
<td>mkdir</td>
<td>228</td>
</tr>
<tr>
<td>multitasking</td>
<td>162, 171</td>
</tr>
<tr>
<td>mutation</td>
<td>261</td>
</tr>
<tr>
<td>mv</td>
<td>9</td>
</tr>
<tr>
<td>NAND gate</td>
<td>142</td>
</tr>
<tr>
<td>natural selection</td>
<td>257</td>
</tr>
<tr>
<td>network layer</td>
<td>190</td>
</tr>
<tr>
<td>Newcomen, Thomas</td>
<td>61</td>
</tr>
<tr>
<td>Nietzsche, Friedrich</td>
<td>286</td>
</tr>
<tr>
<td>node</td>
<td>214</td>
</tr>
<tr>
<td>NOR gate</td>
<td>150</td>
</tr>
<tr>
<td>normalization</td>
<td>252</td>
</tr>
<tr>
<td>NOT gate</td>
<td>142</td>
</tr>
<tr>
<td>NP-complete problem</td>
<td>206</td>
</tr>
<tr>
<td>numerical analysis</td>
<td>202</td>
</tr>
<tr>
<td>object-oriented programming</td>
<td>92, 104</td>
</tr>
<tr>
<td>Occam's razor</td>
<td>131</td>
</tr>
<tr>
<td>opcode</td>
<td>148</td>
</tr>
<tr>
<td>operant conditioning</td>
<td>284</td>
</tr>
<tr>
<td>operating system</td>
<td>5</td>
</tr>
<tr>
<td>option</td>
<td>26</td>
</tr>
<tr>
<td>OR gate</td>
<td>141</td>
</tr>
<tr>
<td>parameter</td>
<td>75, 167</td>
</tr>
<tr>
<td>parser</td>
<td>127</td>
</tr>
<tr>
<td>path</td>
<td>215</td>
</tr>
<tr>
<td>Perl</td>
<td>10, 241</td>
</tr>
<tr>
<td>perl</td>
<td>11</td>
</tr>
<tr>
<td>Perlis, Alan</td>
<td>95</td>
</tr>
<tr>
<td>physical layer</td>
<td>191</td>
</tr>
<tr>
<td>piano-movers problem</td>
<td>233</td>
</tr>
<tr>
<td>PID</td>
<td>39, 170</td>
</tr>
<tr>
<td>pipes in Unix</td>
<td>8, 37</td>
</tr>
<tr>
<td>PLT Scheme</td>
<td>111</td>
</tr>
<tr>
<td>Polish notation</td>
<td>63, 75</td>
</tr>
<tr>
<td>polynomial scaling</td>
<td>203</td>
</tr>
<tr>
<td>POP</td>
<td>164</td>
</tr>
<tr>
<td>port</td>
<td>186</td>
</tr>
<tr>
<td>primary key</td>
<td>49</td>
</tr>
<tr>
<td>probability distribution</td>
<td>1</td>
</tr>
<tr>
<td>process</td>
<td>38</td>
</tr>
<tr>
<td>asynchronous</td>
<td>37, 174</td>
</tr>
<tr>
<td>heavyweight</td>
<td>168</td>
</tr>
<tr>
<td>identifier</td>
<td>39, 170</td>
</tr>
<tr>
<td>parent</td>
<td>38</td>
</tr>
<tr>
<td>thread</td>
<td>168, 172</td>
</tr>
<tr>
<td>processor</td>
<td></td>
</tr>
<tr>
<td>processor clock</td>
<td>170</td>
</tr>
<tr>
<td>program counter</td>
<td>154</td>
</tr>
<tr>
<td>programming environment</td>
<td>12</td>
</tr>
<tr>
<td>Prolog</td>
<td>13, 48</td>
</tr>
<tr>
<td>database</td>
<td>14</td>
</tr>
<tr>
<td>variables</td>
<td>14</td>
</tr>
<tr>
<td>prompt</td>
<td>5</td>
</tr>
<tr>
<td>protocol</td>
<td>185</td>
</tr>
<tr>
<td>ps</td>
<td>38, 178</td>
</tr>
<tr>
<td>pseudo-random number</td>
<td>265</td>
</tr>
<tr>
<td>PUSH</td>
<td>164</td>
</tr>
<tr>
<td>pwd</td>
<td>26</td>
</tr>
<tr>
<td>quasi quote</td>
<td>132</td>
</tr>
<tr>
<td>queue</td>
<td>164</td>
</tr>
<tr>
<td>raisin bread</td>
<td>125</td>
</tr>
<tr>
<td>RAM</td>
<td>153, 181</td>
</tr>
<tr>
<td>random number</td>
<td>265</td>
</tr>
</tbody>
</table>
random-access memory, 153
RCX, 172
recursive definition, 16, 59, 248
redirection, 34
input (<), 36
output (>, 34
register, 150
register machine, 153
regular expression, 43
relational database, see database
relative path name, 6
relay, 144
reverse, 19
rmdir, 228
robot, 155, 172
arm, 232
mobile, 155, 233
program, 157, 172
root, 216
RTS, 165
Scheme, 18, 160, 198, 239
search engine, 239
sed, 8
semantics, 76
semaphore, 177
server, 186
server farm, 239
server process, 186
set!, 89
set-car!, 105
set-vector!, 107
sh, 9, 24
shell, 2, 23
script, 7, 24
secure, 2
variable, 29, 228
sign bit, 148
sink, 231
Sketchpad, 97
Skinner, B. F., 284
sleep, 38
Smalltalk, 94, 95
SMTP, 187
sockets, 40
software engineering, 92
sort, 241
sort, 36
spam, 124
spam filter, 124
specification, 58
SQL, 49, 193
ssh, 2
stack, 164
standard input, 35
standard output, 34
stemming algorithm, 244
stepper, 80
stop word, 242
string, 26
strongly connected graph, 231
structured data, 48
structured query language, 49
subgraph, 216
subroutine, 163
substitution model, 77, 86
subtree, 216
supervised learning, 124
Sussman, Gerald, 97
Sutherland, Ivan, 97
symbolic link, 6, 229
syntactic
saccharin, 115
salt, 115
sugar, 114
syntax, 57
task management, 169
TCP, 189
tcsh, 24
TELNET, 187
term frequency, 245
thread, see process
thumbnail image, 255
time sharing, 121
token, 126
topology, see graph
tour, 208, 258
training examples, 124
training set, 130
transition matrix, 249
transport layer, 189
transpose, 250
traveling salesman problem, 208, 258
tree, 214
truth table, 141
<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turing, Alan</td>
<td>206</td>
</tr>
<tr>
<td>typed variable</td>
<td>64</td>
</tr>
<tr>
<td>UDP</td>
<td>189</td>
</tr>
<tr>
<td>unicode</td>
<td>194</td>
</tr>
<tr>
<td>uniq</td>
<td>36</td>
</tr>
<tr>
<td>universal computer</td>
<td>207, 273</td>
</tr>
<tr>
<td>URL</td>
<td>192, 237</td>
</tr>
<tr>
<td>utilitarianism</td>
<td>279</td>
</tr>
<tr>
<td>vector</td>
<td>225, 244, 250</td>
</tr>
<tr>
<td>vector-ref</td>
<td>107</td>
</tr>
<tr>
<td>vertex</td>
<td>214</td>
</tr>
<tr>
<td>virtual memory</td>
<td>181</td>
</tr>
<tr>
<td>VLSI</td>
<td>145</td>
</tr>
<tr>
<td>von Neumann architecture</td>
<td>153</td>
</tr>
<tr>
<td>von Neumann, John</td>
<td>269</td>
</tr>
<tr>
<td>WAN</td>
<td>215</td>
</tr>
<tr>
<td>wc</td>
<td>4</td>
</tr>
<tr>
<td>whitespace</td>
<td>26</td>
</tr>
<tr>
<td>wildcard</td>
<td>6, 51</td>
</tr>
<tr>
<td>XHTML</td>
<td>192</td>
</tr>
<tr>
<td>XOR gate</td>
<td>143</td>
</tr>
<tr>
<td>yacc</td>
<td>127</td>
</tr>
<tr>
<td>Yeager, Chuck</td>
<td>205</td>
</tr>
<tr>
<td>Zeta Lisp</td>
<td>104</td>
</tr>
</tbody>
</table>