

Spis treści

Przedmowa	9
Uwagi do wydania VI	9
Co odróżnia tę książkę od innych podręczników?	10
Dlaczego C++?	11
Jak należy czytać tę książkę?	11
Co zostało opisane w tej książce?	12
Programy przykładowe	14
Konwencje typograficzne i oznaczenia	15
Rozdział 1. Zanim wystartujemy	17
Czym powinien się charakteryzować algorytm?	18
Jak to wcześniej bywało, czyli wyjątki z historii maszyn algorytmicznych	20
— 1804 —	20
— 1830 i później —	21
— 1890 —	21
— lata 30. XX w. —	21
— lata 40. XX w. —	22
— okres powojenny —	22
— 1969 —	23
— teraz —	23
Jak to się niedawno odbyło, czyli o tym, kto „wymyślił” metodologię programowania	24
Proces koncepcji programów	25
Poziomy abstrakcji opisu i wybór języka	26
Modelowanie działania algorytmów (maszyna Turinga)	28
Poprawność algorytmów	29
Zadania	31
Rozwiązania i wskazówki do zadań	31
Rozdział 2. Rekurencja	33
Definicja rekurencji	33
Ilustracja pojęcia rekurencji	35

Jak wykonują się programy rekurencyjne?	36
Niebezpieczeństwa rekurencji	38
Ciąg Fibonacciego	38
Stack overflow!	40
Pułapek ciąg dalszy	42
Stąd do wieczności	43
Definicja poprawna, ale	43
Typy programów rekurencyjnych	45
Myślenie rekurencyjne	46
Przykład 1. Spirala	47
Przykład 2. Kwadraty „parzyste”	48
Uwagi praktyczne na temat technik rekurencyjnych	50
Zadania	51
Rozwiązania i wskazówki do zadań	53

Rozdział 3. Systemy obliczeniowe i podstawy kodowania **59**

System dziesiętny i kilka definicji	60
System dwójkowy	60
Operacje arytmetyczne na liczbach dwójkowych	61
Operacje logiczne na liczbach dwójkowych	62
Kod BCD	64
System ósemkowy	65
System szesnastkowy	65
Kodowanie liczb ze znakiem	65
Kod znak-moduł (ZM)	66
Kod U2 (system uzupełnienia dwójkowego)	66
Zmienne w pamięci komputera	67
Kodowanie znaków	68
Kodowanie obrazów	70
Mapy bitowe na przykładzie formatu BMP	71

Rozdział 4. Typy i struktury danych **75**

Typy podstawowe i złożone	76
Tablice	77
Ciągi znaków i napisy w C++	78
Typy złożone	80
Struktury i wprowadzenie pojęcia referencji	80
Klasy i programowanie obiektowe	83
Abstrakcyjne struktury danych	83
Listy jednokierunkowe	85
Tablicowa implementacja list	106
Stos	111
Kolejki FIFO	116
Stery i kolejki priorytetowe	119
Drzewa i ich reprezentacje	125
Zbiory	138
STL, czyli struktury danych dla leniuchów	140

Klasyczne kontenery sekwencyjne	141
Adaptery (nakładki na inne kontenery)	147
Kontenery asocjacyjne	148
Algorytmy w STL	151
Dalsze materiały na temat STL	152
Zadania	152
Rozwiązania zadań	153
Rozdział 5. Analiza złożoności algorytmów	155
Definicje i przykłady	156
Jeszcze raz funkcja silnia	160
Zerowanie fragmentu tablicy	163
Wpadamy w pułapkę	165
Różne typy złożoności obliczeniowej	166
Nowe zadanie: uprościć obliczenia!	168
Analiza programów rekurencyjnych	169
Terminologia i definicje	169
Ilustracja metody na przykładzie	170
Rozkład logarytmiczny	171
Przeszukiwanie binarne... tym razem bez matematyki wyższej!	173
Zamiana dziedziny równania rekurencyjnego	174
Funkcja Ackermanna, czyli coś dla smakoszy	174
Złożoność obliczeniowa to nie religia!	176
Techniki optymalizacji programów	176
Zadania	177
Rozwiązania i wskazówki do zadań	178
Rozdział 6. Derekursywacja i optymalizacja algorytmów	181
Jak pracuje kompilator?	182
Odrobina formalizmu nie zaszkodzi!	184
Kilka przykładów derekursywacji algorytmów	185
Derekursywacja z wykorzystaniem stosu	188
Eliminacja zmiennych lokalnych	188
Metoda funkcji przeciwnych	190
Klasyczne schematy derekursywacji	192
Schemat typu while	193
Schemat typu if-else	194
Schemat z podwójnym wywołaniem rekurencyjnym	196
Podsumowanie	198
Rozdział 7. Algorytmy sortowania	199
Sortowanie przez wstawianie, algorytm klasy $O(N^2)$	200
Sortowanie bąbelkowe, algorytm klasy $O(N^2)$	201
Sortowanie szybkie (Quicksort) — algorytm klasy $O(N \log N)$	203
Heapsort — sortowanie przez kopcowanie	206
Scalanie zbiorów posortowanych	209
Sortowanie przez scalanie, algorytm klasy $O(N \log N)$	209

Sortowanie zewnętrzne	211
Uwagi praktyczne	214
Rozdział 8. Algorytmy przeszukiwania	217
Przeszukiwanie liniowe	217
Przeszukiwanie binarne	218
Transformacja kluczowa (hashing)	220
W poszukiwaniu funkcji H	221
Najbardziej znane funkcje H	222
Obsługa konfliktów dostępu	224
Powrót do źródeł	225
Jeszcze raz tablice!	226
Próbkowanie liniowe	226
Podwójne kluczowanie	228
Zastosowania transformacji kluczowej	229
Podsumowanie metod transformacji kluczowej	230
Rozdział 9. Przeszukiwanie tekstów	233
Algorytm typu brute force	233
Nowe algorytmy poszukiwań	235
Algorytm KMP	236
Algorytm Boyera-Moore'a	240
Algorytm Rabina-Karpa	242
Rozdział 10. Zaawansowane techniki programowania	245
Programowanie typu „dziel i zwyciężaj”	246
Odszukiwanie minimum i maksimum w tablicy liczb	247
Mnożenie macierzy o rozmiarze $N \times N$	249
Mnożenie liczb całkowitych	252
Inne znane algorytmy „dziel i zwyciężaj”	253
Algorytmy „żarłoczne”, czyli przekąsić coś nadszedł już czas	253
Problem plecakowy, czyli niełatwe jest życie turysty piechura	254
Wydawanie reszty, czyli „A nie ma pan drobnych?” w praktyce	257
Programowanie dynamiczne	258
Ciąg Fibonacciego	259
Równania z wieloma zmiennymi	260
Najdłuższa wspólna podsekwencja	261
Inne techniki programowania	264
Uwagi bibliograficzne	266
Rozdział 11. Elementy algorytmiki grafów	269
Definicje i pojęcia podstawowe	270
Etykiety i wartości	271
Cykle w grafach	273
Sposoby reprezentacji grafów	276
Reprezentacja tablicowa	276
Słowniki węzłów	278

Listy kontra zbiory	279
Podstawowe operacje na grafach	279
Suma grafów	279
Kompozycja grafów	280
Graf do potęgi	280
Algorytm Roya-Warshalla	281
Algorytm Floyd-Warshalla	284
Algorytm Dijkstry	287
Algorytm Bellmana-Forda	289
Drzewo rozpinające minimalne	289
Algorytm Kruskala	290
Algorytm Prima	291
Przeszukiwanie grafów	291
Strategia „w głąb” (przeszukiwanie zstępujące)	292
Strategia „wszerz”	294
Inne strategie przeszukiwania	295
Problem właściwego doboru	296
Podsumowanie	300
Zadania	300

Rozdział 12. Algorytmy numeryczne **301**

Poszukiwanie miejsc zerowych funkcji	301
Iteracyjne obliczanie wartości funkcji	303
Interpolacja funkcji metodą Lagrange'a	304
Różniczkowanie funkcji	305
Całkowanie funkcji metodą Simpsona	307
Rozwiązywanie układów równań liniowych metodą Gaussa	308
Biblioteka GSL (GNU Scientific Library)	311
Uwagi końcowe	311

Rozdział 13. Czy komputery mogą myśleć? **313**

Przegląd obszarów zainteresowań sztucznej inteligencji (SI)	314
Systemy eksperckie	315
Sieci neuronowe	317
Reprezentacja problemów	318
Gry dwuosobowe i drzewa gier	320
Algorytm min-max	321

Rozdział 14. Kodowanie i kompresja danych **327**

Kodowanie danych i arytmetyka dużych liczb	329
Metody prymitywne	329
Kodowanie symetryczne	331
Kodowanie asymetryczne	332
Łamanie kodów	338
Jakość klucza szyfrującego	338
Metody łamania szyfrów	339
Techniki kompresji danych	340

Kompresja za pomocą modelowania matematycznego	341
Kompresja metodą RLE	342
Kompresja danych metodą Huffmana	343
Kodowanie LZW	348
Rozdział 15. Zadania różne	355
Teksty zadań	355
Rozwiązania	357
Dodatek A Poznaj C++ w pięć minut!	361
Elementy języka C++ na przykładach	361
Pierwszy program	361
Dyrektywa #include	362
Kod warunkowy w C++	362
Operacje arytmetyczne i zmienne	363
Operacje logiczne	363
Wskaźniki i zmienne dynamiczne	364
Referencje	365
Typy proste i typy złożone	365
Podprogramy	367
Procedury	367
Funkcje	367
Instrukcja wyboru (switch)	368
Iteracje	369
Struktury rekurencyjne	369
Parametry programu main()	370
Operacje na plikach w C++	370
Programowanie obiektowe w C++	371
Terminologia	372
Obiekty na przykładzie	373
Składowe statyczne klas	376
Metody stałe klas	376
Dziedziczenie własności	376
Dodatek B Kompilowanie programów przykładowych	381
Zawartość archiwum ZIP na FTP-ie	381
Darmowe kompilatory C++	382
GCC (GNU Compiler Collection)	382
Microsoft Visual Studio Community	384
macOS	386
Dev-C++(Orwell)	386
Kompilacja i uruchamianie programów w C++	387
GCC	387
Microsoft Visual Studio	388
Dev-C++	395
Cygwin	395

Literatura	397
Spis ilustracji	399
Spis tabel	404
Skorowidz	406

oprac. BPK