

Przedmowa (11)

Rozdział 1. Zanim wystartujemy (19)

- 1.1. Jak to wcześniej bywało, czyli wyjątki z historii maszyn algorytmicznych (21)
- 1.2. Jak to się niedawno odbyło, czyli o tym, kto "wymyślił" metodologię programowania (23)
- 1.3. Proces koncepcji programów (24)
- 1.4. Poziomy abstrakcji opisu i wybór języka (25)
- 1.5. Poprawność algorytmów (27)

Rozdział 2. Rekurencja (29)

- 2.1. Definicja rekurencji (29)
- 2.2. Ilustracja pojęcia rekurencji (31)
- 2.3. Jak wykonują się programy rekurencyjne? (32)
- 2.4. Niebezpieczeństwa rekurencji (34)
 - o 2.4.1. Ciąg Fibonacciego (34)
 - o 2.4.2. Stack overflow! (36)
- 2.5. Pułapek ciąg dalszy (37)
 - o 2.5.1. Stąd do wieczności (37)
 - o 2.5.2. Definicja poprawna, ale... (38)
- 2.6. Typy programów rekurencyjnych (39)
- 2.7. Myślenie rekurencyjne (41)
 - o 2.7.1. Przykład 1.: Spirala (42)
 - o 2.7.2. Przykład 2.: Kwadraty "parzyste" (44)
- 2.8. Uwagi praktyczne na temat technik rekurencyjnych (45)
- 2.9. Zadania (46)
 - o 2.9.1. Zadanie 2.1. (46)
 - o 2.9.2. Zadanie 2.2. (46)
 - o 2.9.3. Zadanie 2.3. (48)
 - o 2.9.4. Zadanie 2.4. (48)
 - o 2.9.5. Zadanie 2.5. (49)
- 2.10. Rozwiązania i wskazówki do zadań (49)
 - o 2.10.1. Zadanie 2.1. (49)
 - o 2.10.2. Zadanie 2.2. (50)
 - o 2.10.3. Zadanie 2.3. (50)
 - o 2.10.4. Zadanie 2.4. (51)
 - o 2.10.5. Zadanie 2.5. (52)

Rozdział 3. Analiza sprawności algorytmów (53)

- 3.1. Dobre samopoczucie użytkownika programu (54)
- 3.2. Przykład 1: Jeszcze raz funkcja silnia... (56)
- 3.3. Przykład 2: Zerowanie fragmentu tablicy (60)
- 3.4. Przykład 3: Wpadamy w pułapkę (62)
- 3.5. Przykład 4: Różne typy złożoności obliczeniowej (63)
- 3.6. Nowe zadanie: uprościć obliczenia! (66)
- 3.7. Analiza programów rekurencyjnych (66)
 - o 3.7.1. Terminologia (67)
 - o 3.7.2. Ilustracja metody na przykładzie (68)

- o 3.7.3. Rozkład logarytmiczny (70)
 - o 3.7.4. Zamiana dziedziny równania rekurencyjnego (71)
 - o 3.7.5. Funkcja Ackermanna, czyli coś dla smakoszy (72)
- 3.8. Zadania (73)
 - o 3.8.1. Zadanie 3.1. (73)
 - o 3.8.2. Zadanie 3.2. (74)
 - o 3.8.3. Zadanie 3.3. (74)
 - o 3.8.4. Zadanie 3.4. (74)
- 3.9. Rozwiązania i wskazówki do zadań (75)
 - o 3.9.1. Zadanie 3.2. (75)
 - o 3.9.2. Zadanie 3.4. (76)

Rozdział 4. Algorytmy sortowania (77)

- 4.1. Sortowanie przez wstawianie, algorytm klasy $O(N^2)$ (78)
- 4.2. Sortowanie bąbelkowe, algorytm klasy $O(N^2)$ (80)
- 4.3. Quicksort, algorytm klasy $O(N \log N)$ (82)
- 4.4. Heap Sort - sortowanie przez kopcowanie (85)
- 4.5. Sortowanie przez scalanie (88)
- 4.6. Uwagi praktyczne (90)

Rozdział 5. Struktury danych (93)

- 5.1. Listy jednokierunkowe (94)
 - o 5.1.1. Realizacja struktur danych listy jednokierunkowej (96)
 - o 5.1.2. Tworzenie listy jednokierunkowej (97)
 - o 5.1.3. Listy jednokierunkowe - teoria i rzeczywistość (106)
- 5.2. Tablicowa implementacja list (119)
 - o 5.2.1. Klasyczna reprezentacja tablicowa (119)
 - o 5.2.2. Metoda tablic równoległych (121)
 - o 5.2.3. Listy innych typów (123)
- 5.3. Stos (125)
 - o 5.3.1. Zasada działania stosu (125)
- 5.4. Kolejki FIFO (129)
- 5.5. Sterty i kolejki priorytetowe (132)
- 5.6. Drzewa i ich reprezentacje (139)
 - o 5.6.1. Drzewa binarne i wyrażenia arytmetyczne (142)
- 5.7. Uniwersalna struktura słownikowa (147)
- 5.8. Zbiory (152)
- 5.9. Zadania (155)
 - o 5.9.1. Zadanie 5.1. (155)
 - o 5.9.2. Zadanie 5.2. (155)
 - o 5.9.3. Zadanie 5.3. (155)
- 5.10. Rozwiązania zadań (155)
 - o 5.10.1. Zadanie 5.1. (155)
 - o 5.10.2. Zadanie 5.3. (156)

Rozdział 6. Derekursywacja i optymalizacja algorytmów (157)

- 6.1. Jak pracuje kompilator? (158)

- 6.2. Odrobina formalizmu nie zaszkodzi! (160)
- 6.3. Kilka przykładów derekursywacji algorytmów (162)
- 6.4. Derekursywacja z wykorzystaniem stosu (165)
 - o 6.4.1. Eliminacja zmiennych lokalnych (166)
- 6.5. Metoda funkcji przeciwnych (168)
- 6.6. Klasyczne schematy derekursywacji (170)
 - o 6.6.1. Schemat typu while (171)
 - o 6.6.2. Schemat typu if-else (173)
 - o 6.6.3. Schemat z podwójnym wywołaniem rekurencyjnym (175)
- 6.7. Podsumowanie (177)

Rozdział 7. Algorytmy przeszukiwania (179)

- 7.1. Przeszukiwanie liniowe (179)
- 7.2. Przeszukiwanie binarne (180)
- 7.3. Transformacja kluczowa (hashing) (181)
 - o 7.3.1. W poszukiwaniu funkcji H (183)
 - o 7.3.2. Najbardziej znane funkcje H (184)
 - o 7.3.3. Obsługa konfliktów dostępu (187)
 - o 7.3.4. Powrót do źródeł (187)
 - o 7.3.5. Jeszcze raz tablice! (188)
 - o 7.3.6. Próbkowanie liniowe (189)
 - o 7.3.7. Podwójne kluczowanie (191)
 - o 7.3.8. Zastosowania transformacji kluczowej (193)
 - o 7.3.9. Podsumowanie metod transformacji kluczowej (193)

Rozdział 8. Przeszukiwanie tekstów (195)

- 8.1. Algorytm typu brute-force (195)
- 8.2. Nowe algorytmy poszukiwań (197)
 - o 8.2.1. Algorytm K-M-P (198)
 - o 8.2.2. Algorytm Boyera i Moore'a (203)
 - o 8.2.3. Algorytm Rabina i Karpa (205)

Rozdział 9. Zaawansowane techniki programowania (209)

- 9.1. Programowanie typu "dziel i zwyciężaj" (210)
 - o 9.1.1. Odszukiwanie minimum i maksimum w tablicy liczb (211)
 - o 9.1.2. Mnożenie macierzy o rozmiarze $N(N)$ (214)
 - o 9.1.3. Mnożenie liczb całkowitych (217)
 - o 9.1.4. Inne znane algorytmy "dziel i zwyciężaj" (218)
- 9.2. Algorytmy "żarłoczne", czyli przekąsić coś nadszedł już czas... (219)
 - o 9.2.1. Problem plecakowy, czyli niełatwe jest życie turysty-piechura (220)
- 9.3. Programowanie dynamiczne (223)
- 9.4. Uwagi bibliograficzne (227)

Rozdział 10. Elementy algorytmiki grafów (229)

- 10.1. Definicje i pojęcia podstawowe (230)
- 10.2. Sposoby reprezentacji grafów (232)

- 10.3. Podstawowe operacje na grafach (234)
 - o 10.3.1. Suma grafów (234)
 - o 10.3.2. Kompozycja grafów (234)
 - o 10.3.3. Potęga grafu (235)
- 10.4. Algorytm Roy-Warshalla (236)
- 10.5. Algorytm Floyda-Warshalla (239)
- 10.6. Algorytm Dijkstry (243)
- 10.7. Drzewo rozpinające minimalne (244)
- 10.8. Przeszukiwanie grafów (245)
 - o 10.8.1. Strategia "w głąb" (246)
 - o 10.8.2. Strategia "wszerz" (247)
- 10.9. Problem właściwego doboru (249)
- 10.10. Podsumowanie (254)

Rozdział 11. Algorytmy numeryczne (255)

- 11.1. Poszukiwanie miejsc zerowych funkcji (255)
- 11.2. Iteracyjne obliczanie wartości funkcji (257)
- 11.3. Interpolacja funkcji metodą Lagrange'a (258)
- 11.4. Różniczkowanie funkcji (260)
- 11.5. Całkowanie funkcji metodą Simpsona (262)
- 11.6. Rozwiązywanie układów równań liniowych metodą Gaussa (263)
- 11.7. Uwagi końcowe (266)

Rozdział 12. Czy komputery mogą myśleć? (267)

- 12.1. Przegląd obszarów zainteresowań Sztucznej Inteligencji (268)
 - o 12.1.1. Systemy eksperckie (269)
 - o 12.1.2. Sieci neuronowe (271)
- 12.2. Reprezentacja problemów (272)
 - o 12.2.1. Ćwiczenie 12.1. (274)
- 12.3. Gry dwuosobowe i drzewa gier (274)
- 12.4. Algorytm mini-max (276)

Rozdział 13. Kodowanie i kompresja danych (281)

- 13.1. Kodowanie danych i arytmetyka dużych liczb (283)
 - o 13.1.1. Kodowanie symetryczne (284)
 - o 13.1.2. Kodowanie asymetryczne (285)
 - o 13.1.3. Metody prymitywne (292)
 - o 13.1.4. Łamanie szyfrów (293)
- 13.2. Techniki kompresji danych (294)
 - o 13.2.1. Kompresja poprzez modelowanie matematyczne (296)
 - o 13.2.2. Kompresja metodą RLE (297)
 - o 13.2.3. Kompresja danych metodą Huffmana (298)
 - o 13.2.4. Kodowanie LZW (304)
 - o 13.2.5. Idea kodowania słownikowego na przykładach (304)
 - o 13.2.6. Opis formatu GIF (307)

Rozdział 14. Zadania różne (311)

- 14.1. Teksty zadań (311)
 - o 14.1.1. Zadanie 14.1. (311)
 - o 14.1.2. Zadanie 14.2. (312)
 - o 14.1.3. Zadanie 14.3. (312)
 - o 14.1.4. Zadanie 14.4. (312)
 - o 14.1.5. Zadanie 14.5. (312)
 - o 14.1.6. Zadanie 14.6. (312)
 - o 14.1.7. Zadanie 14.7. (312)
 - o 14.1.8. Zadanie 14.8. (313)
 - o 14.1.9. Zadanie 14.9. (313)
 - o 14.1.10. Zadanie 14.10. (313)
 - o 14.1.11. Zadanie 14.11. (313)
 - o 14.1.12. Zadanie 14.12. (313)
- 14.2. Rozwiązania (313)
 - o 14.2.1. Zadanie 14.1. (313)
 - o 14.2.2. Zadanie 14.3. (314)
 - o 14.2.3. Zadanie 14.4. (315)
 - o 14.2.4. Zadanie 14.10. (315)
 - o 14.2.5. Zadanie 14.12. (316)

Dodatek A Poznaj C++ w pięć minut! (319)

Dodatek B Systemy obliczeniowe w pigułce (337)

Literatura (343)

Spis ilustracji (345)

Spis tablic (349)

Skorowidz (351)