

O Autorze (19)

Drogi Czytelniku (20)

Wprowadzenie (21)

- Co nowego w drugim wydaniu? (21)
 - Dodatkowe tematy (21)
 - Pytania końcowe (22)
 - Eksperymenty (22)
 - Projekty programistyczne (22)
- O czym jest ta książka? (22)
- Czym ta książka różni się od innych? (23)
 - Łatwa do zrozumienia (23)
 - Warsztaty algorytmów (24)
 - Przykłady w Javie (24)
- Komu może się przydać ta książka? (25)
- Co musimy wiedzieć, zanim zaczniemy ją czytać? (25)
- Potrzebne oprogramowanie (25)
- Organizacja książki (25)
- Dobrej zabawy! (27)

1. Przegląd (29)

- Do czego są przydatne struktury danych i algorytmy? (29)
 - Przechowywanie danych ze świata zewnętrznego (30)
 - Narzędzia programisty (31)
 - Modelowanie danych ze świata zewnętrznego (31)
- Przegląd struktur danych (31)
- Przegląd algorytmów (31)
- Kilka definicji (32)
 - Baza danych (32)
 - Rekord (33)
 - Pole (33)
 - Klucz (33)
- Programowanie obiektowe (34)
 - Problemy z językami proceduralnymi (34)
 - Obiekty w telegraficznym skrócie (35)
 - Działający program obiektowy (37)
 - Dziedziczenie i polimorfizm (39)
- Inżynieria programowania (40)
- Java dla programistów C++ (40)
 - Brak wskaźników (41)
 - Operatory przeciążone (44)
 - Typy proste (44)
 - Operacje wejścia-wyjścia (44)
- Struktury danych w bibliotece Javy (47)
- Podsumowanie (47)
- Pytania (48)

2. Tablice (49)

- Aplet demonstracyjny (49)
 - Wstawianie (51)
 - Wyszukiwanie (51)
 - Usuwanie (52)
 - Kwestia duplikatów (53)
 - Niezbyt szybko (55)
- Tablice w Javie (55)
 - Tworzenie tablicy (55)
 - Dostęp do elementów tablicy (56)
 - Inicjalizacja (56)
 - Przykład użycia tablic (57)
- Dzielenie programu na klasy (59)
 - Klasy LowArray i LowArrayApp (61)
- Interfejsy klas (61)
 - Niezbyt wygodnie (62)
 - Kto i za co odpowiada? (62)
 - Program highArray.java (63)
 - ... i życie użytkownika stało się prostsze (66)
 - Abstrakcja (66)
- Aplet demonstrujący tablicę uporządkowaną (66)
 - Wyszukiwanie liniowe (66)
 - Wyszukiwanie binarne (67)
- Tablica uporządkowana w Javie (69)
 - Wyszukiwanie binarne w metodzie find() (70)
 - Klasa OrdArray (71)
 - Korzyści wynikające z używania tablic uporządkowanych (74)
- Logarytmy (74)
 - Potęgowanie (75)
 - Przeciwnieństwo podnoszenia do potęgi (76)
- Przechowywanie obiektów (76)
 - Klasa Person (76)
 - Program classDataArray.java (77)
- Notacja O() (81)
 - Wstawianie do tablicy nieuporządkowanej: czas stały (81)
 - Wyszukiwanie liniowe: czas proporcjonalny do N (81)
 - Wyszukiwanie binarne: czas proporcjonalny do log(N) (82)
 - Stała niepotrzebna (82)
- Czy tablice nadają się do wszystkiego? (83)
- Podsumowanie (84)
- Pytania (85)
- Eksperymenty (86)
- Projekty programistyczne (86)

3. Proste algorytmy sortowania (87)

- W szeregu zbiórka! (88)
- Sortowanie bąbelkowe (89)
 - Sortowanie bąbelkowe zawodników drużyny (89)
 - Aplet demonstracyjny sortowania bąbelkowego (91)
 - Sortowanie bąbelkowe w Javie (94)

- Niezmienniki (97)
 - Wydajność sortowania bąbelkowego (97)
- Sortowanie przez wybór (98)
 - Sortowanie przez wybór drużyny baseballowej (98)
 - Aplet demonstracyjny sortowania przez wybór (100)
 - Sortowanie przez wybór w Javie (101)
 - Niezmiennik (103)
 - Wydajność sortowania przez wybór (103)
- Sortowanie przez wstawianie (103)
 - Sortowanie przez wstawianie drużyny baseballowej (103)
 - Aplet demonstracyjny sortowania przez wstawianie (104)
 - Sortowanie przez wstawianie w Javie (107)
 - Niezmiennik w sortowaniu przez wstawianie (110)
 - Wydajność sortowania przez wstawianie (110)
- Sortowanie obiektów (111)
 - Program sortujący tablicę obiektów (111)
 - Porównania leksykograficzne (114)
 - Stabilność (115)
- Porównanie prostych algorytmów sortowania (115)
- Podsumowanie (115)
- Pytania (116)
- Eksperymenty (117)
- Projekty programistyczne (118)

4. Stosy i kolejki (121)

- Inny rodzaj struktur danych (121)
 - Narzędzia programisty (121)
 - Ograniczony dostęp (122)
 - Bardziej abstrakcyjne (122)
- Stosy (122)
 - Analogia pocztowa (123)
 - Aplet demonstracyjny stosu (124)
 - Stos w Javie (126)
 - Wykorzystanie stosu do odwracania słowa (129)
 - Wykorzystanie stosu do sprawdzania nawiasów (131)
 - Wydajność stosów (136)
- Kolejki (136)
 - Aplet demonstracyjny kolejki (137)
 - Kolejka cykliczna (140)
 - Kolejka w Javie (141)
 - Wydajność kolejek (146)
 - Kolejki dwustronne (146)
- Kolejki priorytetowe (146)
 - Aplet demonstracyjny kolejki priorytetowej (147)
 - Kolejka priorytetowa w Javie (150)
 - Wydajność kolejek priorytetowych (152)
- Analiza wyrażeń arytmetycznych (152)
 - Notacja przyrostkowa (152)
 - Zamiana notacji naturalnej na przyrostkową (153)

- Obliczanie wyrażeń w notacji przyrostkowej (167)
- Podsumowanie (172)
- Pytania (172)
- Eksperymenty (174)
- Projekty programistyczne (174)

5. Listy powiązane (177)

- Połączenia (178)
 - Referencje i typy proste (179)
 - Relacja, nie pozycja (180)
- Aplet demonstracyjny listy powiązanej (181)
 - Przycisk Ins (181)
 - Przycisk Find (182)
 - Przycisk Del (182)
- Prosta lista powiązana (183)
 - Klasa Link (183)
 - Klasa LinkList (184)
 - Metoda insertFirst() (185)
 - Metoda deleteFirst() (186)
 - Metoda displayList() (187)
 - Program linkList.java (187)
- Wyszukiwanie i usuwanie określonych elementów (190)
 - Metoda find() (193)
 - Metoda delete() (193)
 - Inne metody (194)
- Listy dwustronne (194)
- Wydajność list powiązanych (198)
- Abstrakcyjne typy danych (199)
 - Implementacja stosu przy użyciu listy powiązanej (199)
 - Implementacja kolejki przy użyciu listy powiązanej (202)
 - Typy danych i abstrakcja (205)
 - Listy abstrakcyjne (206)
 - Abstrakcyjne typy danych jako narzędzia projektowe (206)
- Listy uporządkowane (207)
 - Wstawianie do listy uporządkowanej w Javie (209)
 - Program sortedList.java (210)
 - Wydajność list uporządkowanych (212)
 - Sortowanie przez wstawianie do listy (212)
- Listy dwukierunkowe (214)
 - Przeglądanie (216)
 - Wstawianie (216)
 - Usuwanie (218)
 - Program doublyLinked.java (219)
 - Listy dwukierunkowe jako podstawa kolejek dwustronnych (223)
- Iteratory (223)
 - Referencja w liście? (224)
 - Klasa iteratora (224)
 - Dodatkowe możliwości iteratorów (225)
 - Metody klasy iteratorowej (226)

- Program interIterator.java (227)
- Na co wskazuje iterator? (232)
- Metoda atEnd() (233)
- Operacje iteracyjne (233)
- Inne metody (234)
- Podsumowanie (235)
- Pytania (236)
- Eksperymenty (237)
- Projekty programistyczne (237)

6. Rekurencja (239)

- Liczby trójkątne (239)
 - Określanie wartości n-tego elementu przy użyciu pętli (240)
 - Określanie wartości n-tego elementu przy użyciu rekurencji (241)
 - Program triangle.java (243)
 - Co się tak naprawdę dzieje? (244)
 - Charakterystyczne cechy metod rekurencyjnych (245)
 - Czy rekurencja jest efektywna? (246)
 - Indukcja matematyczna (247)
- Silnia (247)
- Anagramy (248)
- Rekurencyjne wyszukiwanie binarne (254)
 - Zastąpienie pętli rozwiązaniem rekurencyjnym (255)
 - Algorytmy "dziel i zwyciężaj" (258)
- Wieże Hanoi (259)
 - Aplet Towers Workshop (260)
 - Przesuwanie poddrzew (261)
 - Algorytm rekurencyjny (262)
 - Program towers.java (262)
- Sortowanie przez scalanie (265)
 - Scalanie dwóch posortowanych tablic (265)
 - Sortowanie przez scalanie (268)
 - Aplet MergeSort Workshop (271)
 - Program mergeSort.java (272)
 - Efektywność działania algorytmu sortowania przez scalanie (276)
- Eliminacja rekurencji (278)
 - Rekurencja i stosy (279)
 - Symulowanie metod rekurencyjnych (279)
 - Czego to dowodzi? (284)
- Niektóre interesujące zastosowania rekurencji (286)
 - Podnoszenie liczby do potęgi (287)
 - Problem plecakowy (288)
 - Kombinacje: Wybieranie zespołu (290)
- Podsumowanie (292)
- Pytania (293)
- Eksperymenty (294)
- Projekty programów (294)

7. Zaawansowane algorytmy sortowania (297)

- Sortowanie Shella (297)
 - Sortowanie przez wstawianie: zbyt wiele operacji kopiowania (298)
 - N-sortowanie (298)
 - Usuwanie odstępów (300)
 - Aplet Shellsort Workshop (301)
 - Kod algorytmu Shella napisany w Javie (303)
 - Inne sekwencje dostępu (306)
 - Efektywność działania algorytmu Shella (306)
- Podział danych (307)
 - Aplet demonstracyjny Partitioning (307)
 - Program Partition.java (309)
 - Algorytm podziału danych (311)
 - Efektywność działania algorytmu podziału (314)
- Quicksort (314)
 - Algorytm quicksort (315)
 - Wybór wartości osiowej (316)
 - Aplet demonstracyjny QuickSort1 (321)
 - Obniżenie efektywności do rzędu $O(N^2)$ (325)
 - Wybór mediany trzech elementów (326)
 - Obsługa dzielenia niewielkich grup danych (331)
 - Usuwanie rekurencji (335)
 - Efektywność działania algorytmu quicksort (335)
- Sortowanie pozycyjne (338)
 - Algorytm sortowania pozycyjnego (338)
 - Projekt programu (339)
 - Efektywność sortowania pozycyjnego (339)
- Podsumowanie (340)
- Pytania (341)
- Eksperymenty (343)
- Projekty programów (343)

8. Drzewa binarne (345)

- Dlaczego warto używać drzew binarnych? (345)
 - Wolne wstawianie elementów do tablicy uporządkowanych (346)
 - Wolne wyszukiwanie w listach powiązanych (346)
 - Rozwiązaniem są drzewa (347)
 - Czym jest drzewo? (347)
- Terminologia związana z drzewami (348)
 - Ścieżka (348)
 - Korzeń (348)
 - Rodzic (349)
 - Potomek (349)
 - Liście (349)
 - Poddzewo (350)
 - Odwiedzanie (350)
 - Trawersowanie (350)
 - Poziomy (350)
 - Klucze (350)
 - Drzewa binarne (350)

- Analogia (351)
- Jak działają drzewa binarne? (352)
 - Aplet demonstracyjny Binary Tree (352)
 - Reprezentacja drzew w języku Java (354)
- Wyszukiwanie węzła (357)
 - Wyszukiwanie węzłów w aplecie demonstracyjnym Binary Tree (357)
 - Kod metody wyszukującej węzeł (358)
 - Efektywność operacji na drzewach binarnych (359)
- Wstawianie węzła (359)
 - Wstawianie węzłów w aplecie demonstracyjnym Binary Tree (359)
 - Kod metody wstawiającej węzeł (360)
- Trawersowanie drzewa (362)
 - Trawersowanie drzew w porządku inorder (362)
 - Kod metody trawersującej drzewo (362)
 - Trawersowanie drzew zawierających trzy węzły (363)
 - Trawersowanie drzewa w aplecie demonstracyjnym Binary Tree (364)
 - Trawersowanie drzew w porządkach preorder oraz postorder (366)
- Znajdowanie wartości maksymalnej i minimalnej (368)
- Usuwanie węzła (369)
 - Przypadek 1. Usuwany węzeł nie ma potomków (370)
 - Przypadek 2. Usuwany węzeł ma jednego potomka (372)
 - Przypadek 3. Usuwany węzeł ma dwa potomki (373)
- Efektywność operacji na drzewach binarnych (381)
- Przedstawianie drzew w formie tablicy (383)
- Powtarzające się klucze (384)
- Program tree.java (385)
- Kod Huffmana (393)
 - Kody znaków (393)
 - Dekodowanie przy wykorzystaniu drzewa Huffmana (395)
 - Tworzenie drzewa Huffmana (396)
 - Kodowanie tekstu wiadomości (397)
 - Tworzenie kodów Huffmana (398)
- Podsumowanie (399)
- Pytania (401)
- Eksperymenty (402)
- Projekty programów (402)

9. Drzewa czerwono-czarne (405)

- Sposób omówienia struktury (406)
 - Zasada działania (406)
 - Wstawianie zstępujące (406)
- Drzewa zrównoważone i drzewa niezrównoważone (406)
 - Degeneracja do $O(N)$ (407)
 - Równoważenie drzewa (408)
 - Cechy drzewa czerwono-czarnego (408)
 - Korygowanie struktury (410)
- Aplet demonstracyjny RBTree (410)
 - Kliknięcie obrazka węzła (411)
 - Przycisk Start (411)

- Przycisk Ins (411)
- Przycisk Del (411)
- Przycisk Flip (411)
- Przycisk RoL (412)
- Przycisk RoR (412)
- Przycisk R/B (412)
- Komunikaty tekstowe (412)
- Gdzie jest przycisk Find? (412)
- Ćwiczenia z apilem demonstracyjnym (413)
 - Ćwiczenie 2. - obroty (414)
 - Ćwiczenie 3. - odwracanie kolorów (414)
 - Ćwiczenie 4. - drzewo niezrównoważone (415)
 - Dalsze ćwiczenia (416)
 - Reguły RB i drzewa zrównoważone (416)
 - Potomek pusty (416)
- Obroty (417)
 - Proste operacje obrotu (417)
 - Tajemniczy węzeł krzyżowy (418)
 - Obracanie gałęzi drzewa (418)
 - Ludzie przeciw komputerom (420)
- Wstawianie nowego węzła (421)
 - Przebieg procedury wstawiania (421)
 - Odwrócenia kolorów w trakcie przeszukiwania (422)
 - Obroty po wstawieniu węzła (423)
 - Obroty w trakcie przeszukiwania (429)
- Usuwanie (432)
- Wydajność drzew czerwono-czarnych (432)
- Implementacja drzewa czerwono-czarnego (433)
- Inne drzewa zrównoważone (433)
- Podsumowanie (434)
- Pytania (434)
- Ćwiczenia (436)

10. Drzewa 2-3-4 i pamięć zewnętrzna (437)

- Wprowadzenie (437)
 - Skąd nazwa? (438)
 - Organizacja drzewa 2-3-4 (439)
 - Przeszukiwanie drzewa 2-3-4 (440)
 - Wstawianie danych (440)
 - Podziały węzłów (441)
 - Podział korzenia (442)
 - Zstępujące dzielenie węzłów (442)
- Aplet demonstracyjny Tree234 (443)
 - Przycisk Fill (443)
 - Przycisk Find (444)
 - Przycisk Ins (445)
 - Przycisk Zoom (445)
 - Przeglądanie węzłów (446)
 - Ćwiczenia (447)

- Kod drzewa 2-3-4 w języku Java (448)
 - Klasa DataItem (449)
 - Klasa Node (449)
 - Klasa Tree234 (449)
 - Klasa Tree234App (450)
 - Pełny kod programu tree234.java (451)
- Drzewa 2-3-4 a drzewa czerwono-czarne (458)
 - Transformacja drzewa 2-3-4 do drzewa czerwono-czarnego (458)
 - Równoważność operacji (460)
- Wydajność drzew 2-3-4 (461)
 - Szybkość (461)
 - Wymagania pamięciowe (463)
- Drzewa 2-3 (463)
 - Podziały węzłów (464)
 - Implementacja (466)
- Pamięć zewnętrzna (466)
 - Dostęp do danych zewnętrznych (467)
 - Sekwencyjne porządkowanie danych (470)
 - B-drzewa (471)
 - Indeksowanie (476)
 - Złożone kryteria wyszukiwania (479)
 - Sortowanie plików zewnętrznych (479)
- Podsumowanie (481)
- Pytania (483)
- Ćwiczenia (484)
- Propozycje programów (485)

11. Tablice rozproszone (487)

- Algorytmy rozpraszania - wprowadzenie (488)
 - Numery pracowników jako klucze danych (488)
 - Słownik (489)
 - Rozpraszanie (492)
 - Kolizje (494)
- Adresowanie otwarte (495)
 - Aplet demonstracyjny Hash (496)
 - Kod tablicy rozproszonej z sondowaniem liniowym (500)
 - Sondowanie kwadratowe (507)
 - Podwójne rozpraszanie (510)
- Łączenie niezależne (516)
 - Aplet demonstracyjny HashChain (517)
 - Kod łączenia niezależnego w języku Java (519)
- Funkcje rozpraszające (524)
 - Szybkie obliczanie wyniku (524)
 - Losowe wartości kluczy (525)
 - Nielosowe wartości kluczy (525)
 - Funkcje rozpraszające ciągów znakowych (526)
 - Składanie (528)
- Wydajność tablic rozproszonych (529)
 - Adresowanie otwarte (529)

- Łączenie niezależne (531)
 - Adresowanie otwarte a wiązanie niezależne (533)
- Algorytmy rozpraszania i pamięć zewnętrzna (533)
 - Tablica wskaźników do pliku (534)
 - Bloki częściowo wypełnione (534)
 - Bloki pełne (534)
- Podsumowanie (535)
- Pytania (537)
- Ćwiczenia (538)
- Propozycje programów (538)

12. Sterty (541)

- Wprowadzenie (542)
 - Kolejki priorytetowe, sterty i abstrakcyjne typy danych (542)
 - Słabe uporządkowanie (544)
 - Usuwanie danych (544)
 - Wstawianie danych (546)
 - Operacja zamiany węzłów (546)
- Aplet demonstracyjny Heap (548)
 - Przycisk Fill (548)
 - Przycisk Chng (549)
 - Przycisk Rem (549)
 - Przycisk Ins (549)
- Przykład implementacji sterty (549)
 - Wstawianie danych (550)
 - Usuwanie danych (551)
 - Zmiana klucza (552)
 - Rozmiar tablicy (553)
 - Program heap.java (553)
 - Zwiększanie rozmiaru tablicy sterty (558)
 - Wydajność operacji na stercie (559)
- Sterta oparta na drzewie (559)
- Sortowanie stertowe (560)
 - Opuszczanie węzłów (561)
 - Użycie tej samej tablicy (563)
 - Program heapSort.java (563)
 - Wydajność sortowania stertowego (568)
- Podsumowanie (568)
- Pytania (569)
- Ćwiczenia (570)
- Propozycje programów (570)

13. Grafy (573)

- Wprowadzenie (573)
 - Definicje (574)
 - Nota historyczna (576)
 - Reprezentacja grafu w programie (577)
 - Wstawianie wierzchołków i krawędzi (579)

- Klasa Graph (580)
- Wyszukiwanie (581)
 - Wyszukiwanie "wglęb" (DFS) (582)
 - Przeszukiwanie "wszerz" (BFS) (591)
- Minimalne drzewo rozpinające (597)
 - Aplet demonstracyjny GraphN (598)
 - Kod algorytmu minimalnego drzewa rozpinającego (599)
 - Program mst.java (600)
- Sortowanie topologiczne grafów skierowanych (603)
 - Przykład - warunki wstępne kursów (604)
 - Grafy skierowane (604)
 - Sortowanie topologiczne (605)
 - Aplet demonstracyjny GraphD (606)
 - Cykle i drzewa (607)
 - Kod algorytmu sortowania topologicznego (608)
- Spójność w grafach skierowanych (613)
 - Tabela połączeń (614)
 - Algorytm Warshalla (614)
 - Implementacja algorytmu Warshalla (617)
- Podsumowanie (617)
- Pytania (617)
- Ćwiczenia (618)
- Propozycje programów (619)

14. Grafy ważone (621)

- Minimalne drzewo rozpinające grafu ważonego (621)
 - Przykład: telewizja kablowa w dżungli (622)
 - Aplet demonstracyjny GraphW (622)
 - Wysyłamy inspektorów (623)
 - Algorytm (627)
 - Kod algorytmu (629)
 - Program mstw.java (631)
- Problem najkrótszej ścieżki (636)
 - Linia kolejowa (636)
 - Algorytm Dijkstry (638)
 - Agenci i podróże pociągami (638)
 - Aplet demonstracyjny GraphDW (642)
 - Kod algorytmu (646)
 - Program path.java (650)
- Problem najkrótszej ścieżki dla wszystkich par wierzchołków (654)
- Wydajność (656)
- Problemy nierozwiązywalne (657)
 - Wędrowka Skoczka (657)
 - Problem komiwojażera (658)
 - Cykle Hamiltona (658)
- Podsumowanie (659)
- Pytania (659)
- Ćwiczenia (660)
- Propozycje programów (661)

15. Właściwe stosowanie struktur i algorytmów (663)

- Uniwersalne struktury danych (663)
 - Szybkość pracy i algorytmy (664)
 - Biblioteki (665)
 - Tablice (666)
 - Listy powiązane (666)
 - Drzewa przeszukiwań binarnych (666)
 - Drzewa zrównoważone (667)
 - Tablice rozproszone (667)
 - Porównanie uniwersalnych struktur danych (668)
- Wyszczególnione struktury danych (668)
 - Stos (669)
 - Kolejka (669)
 - Kolejka priorytetowa (669)
 - Porównanie wyszczególnionych struktur danych (670)
- Sortowanie (670)
- Grafy (671)
- Pamięć zewnętrzna (671)
 - Zapis sekwencyjny (672)
 - Pliki indeksowane (672)
 - B-drzewa (672)
 - Algorytmy rozpraszania (672)
 - Pamięć wirtualna (673)
- Co dalej? (674)

Dodatek A Uruchamianie apletów demonstracyjnych i programów przykładowych (675)

- Aplety demonstracyjne (675)
- Programy przykładowe (676)
- Software Development Kit firmy Sun Microsystems (676)
 - Wiersz poleceń (676)
 - Ustawienie ścieżki (677)
 - Wyświetlanie apletów demonstracyjnych (677)
 - Praca z apletami demonstracyjnymi (678)
 - Uruchamianie przykładowych programów (678)
 - Kompilowanie programów przykładowych (679)
 - Modyfikowanie kodu źródłowego (679)
 - Kończenie pracy programów przykładowych (679)
- Pliki klas (679)
- Inne systemy wspomaganie programowania (680)

Dodatek B Literatura (681)

- Algorytmy i struktury danych (681)
- Obiektowe języki programowania (682)
- Obiektowe projektowanie i inżynieria oprogramowania (682)

Dodatek C Odpowiedzi na pytania sprawdzające (683)

- Rozdział 1., "Przegląd" (683)
 - Odpowiedzi na pytania (683)
- Rozdział 2., "Tablice" (684)
 - Odpowiedzi na pytania (684)
- Rozdział 3., "Proste algorytmy sortowania" (684)
 - Odpowiedzi na pytania (684)
- Rozdział 4., "Stosy i kolejki" (685)
 - Odpowiedzi na pytania (685)
- Rozdział 5., "Listy powiązane" (685)
 - Odpowiedzi na pytania (685)
- Rozdział 6., "Rekurencja" (686)
 - Odpowiedzi na pytania (686)
- Rozdział 7., "Zaawansowane algorytmy sortowania" (686)
 - Odpowiedzi na pytania (686)
- Rozdział 8., "Drzewa binarne" (687)
 - Odpowiedzi na pytania (687)
- Rozdział 9., "Drzewa czerwono-czarne" (687)
 - Odpowiedzi na pytania (687)
- Rozdział 10., "Drzewa 2-3-4 i pamięć zewnętrzna" (688)
 - Odpowiedzi na pytania (688)
- Rozdział 11., "Tablice rozproszone" (688)
 - Odpowiedzi na pytania (688)
- Rozdział 12., "Sterty" (689)
 - Odpowiedzi na pytania (689)
- Rozdział 13., "Grafy" (689)
 - Odpowiedzi na pytania (689)
- Rozdział 14., "Grafy ważone" (689)
 - Odpowiedzi na pytania (689)

Skorowidz (691)