

## **O autorach (9)**

## **Podziękowania (11)**

## **Wprowadzenie (13)**

## **Rozdział 1. Zaczynamy (23)**

- Czym są algorytmy? (23)
- Co to jest złożoność algorytmu? (26)
- Porównywanie złożoności i notacja "dużego O" (27)
  - Złożoność stała -  $O(1)$  (29)
  - Złożoność liniowa -  $O(N)$  (29)
  - Złożoność kwadratowa -  $O(N^2)$  (30)
  - Złożoność logarytmiczna -  $O(\log N)$  i  $O(N \log N)$  (31)
  - Złożoność rzędu silni -  $O(N!)$  (32)
- Testowanie modułów (32)
  - Czym jest testowanie modułów? (33)
  - Dlaczego testowanie modułów jest ważne? (35)
  - Biblioteka JUnit i jej wykorzystywanie (35)
  - Programowanie sterowane testami (38)
- Podsumowanie (39)

## **Rozdział 2. Iteracja i rekurencja (41)**

- Wykonywanie obliczeń (42)
- Przetwarzanie tablic (44)
  - Iteratory jako uogólnienie przetwarzania tablicowego (45)
- Rekurencja (62)
  - Przykład - rekurencyjne drukowanie drzewa katalogów (64)
  - Anatomia algorytmu rekurencyjnego (68)
- Podsumowanie (69)
- Ćwiczenia (70)

## **Rozdział 3. Listy (71)**

- Czym są listy? (71)
- Testowanie list (74)
- Implementowanie list (86)
  - Lista tablicowa (87)
  - Lista wiązana (95)
- Podsumowanie (104)
- Ćwiczenia (104)

## **Rozdział 4. Kolejki (105)**

- Czym są kolejki? (105)
  - Operacje kolejkowe (106)
  - Interfejs kolejki (107)
- Kolejka FIFO (107)
  - Implementowanie kolejki FIFO (111)
- Kolejki blokujące (113)
- Przykład - symulacja centrum obsługi (117)

- o Uruchomienie aplikacji (127)
- Podsumowanie (128)
- Ćwiczenia (129)

## **Rozdział 5. Stosy (131)**

- Czym są stosy? (131)
- Testy (133)
- Implementacja (136)
- Przykład - implementacja operacji "Cofnij/Powtórz" (140)
  - o Testowanie cofania/powtarzania (141)
- Podsumowanie (149)

## **Rozdział 6. Sortowanie - proste algorytmy (151)**

- Znaczenie sortowania (151)
- Podstawy sortowania (152)
- Komparatory (153)
  - o Operacje komparatora (153)
  - o Interfejs komparatora (154)
  - o Niektóre komparatory standardowe (154)
- Sortowanie bąbelkowe (159)
  - o Interfejs ListSorter (161)
  - o Abstrakcyjna klasa testowa dla sortowania list (161)
- Sortowanie przez wybieranie (165)
- Sortowanie przez wstawianie (170)
- Stabilność sortowania (173)
- Porównanie prostych algorytmów sortowania (175)
  - o CallCountingListComparator (176)
  - o ListSorterCallCountingTest (177)
  - o Jak interpretować wyniki tej analizy? (180)
- Podsumowanie (180)
- Ćwiczenia (181)

## **Rozdział 7. Sortowanie zaawansowane (183)**

- Sortowanie metodą Shella (183)
- Sortowanie szybkie (189)
- Komparator złożony i jego rola w zachowaniu stabilności sortowania (195)
- Sortowanie przez łączenie (198)
  - o Łączenie list (198)
  - o Algorytm Mergesort (199)
- Porównanie zaawansowanych algorytmów sortowania (205)
- Podsumowanie (208)
- Ćwiczenia (209)

## **Rozdział 8. Kolejki priorytetowe (211)**

- Kolejki priorytetowe (212)
  - o Prosty przykład kolejki priorytetowej (212)

- Wykorzystywanie kolejek priorytetowych (215)
  - Kolejka priorytetowa oparta na liście nieposortowanej (218)
  - Kolejka priorytetowa wykorzystująca listę posortowaną (220)
  - Kolejki priorytetowe o organizacji stogowej (222)
- Porównanie implementacji kolejek priorytetowych (229)
- Podsumowanie (233)
- Ćwiczenia (233)

## **Rozdział 9. Binarne wyszukiwanie i wstawianie (235)**

- Wyszukiwanie binarne (235)
  - Dwa sposoby realizacji wyszukiwania binarnego (238)
  - Interfejs wyszukiwania binarnego (238)
  - Iteracyjna wyszukiwarka binarna (244)
  - Ocena działania wyszukiwarek (247)
- Wstawianie binarne (253)
  - Inserter binarny (254)
  - Porównywanie wydajności (257)
- Podsumowanie (261)

## **Rozdział 10. Binarne drzewa wyszukiwawcze (263)**

- Binarne drzewa wyszukiwawcze (264)
  - Minimum (265)
  - Maksimum (265)
  - Następnik (265)
  - Poprzednik (266)
  - Szukanie (266)
  - Wstawianie (268)
  - Usuwanie (269)
  - Trawersacja in-order (272)
  - Trawersacja pre-order (272)
  - Trawersacja post-order (273)
  - Wyważanie drzewa (273)
- Testowanie i implementowanie binarnych drzew wyszukiwawczych (275)
- Ocena efektywności binarnego drzewa wyszukiwawczego (299)
- Podsumowanie (302)
- Ćwiczenia (303)

## **Rozdział 11. Haszowanie (305)**

- Podstawy haszowania (305)
- Praktyczna realizacja haszowania (311)
  - Próbkowanie liniowe (314)
  - Porcjowanie (321)
- Ocena efektywności tablic haszowanych (326)
- Podsumowanie (332)
- Ćwiczenia (332)

## **Rozdział 12. Zbiory (333)**

- Podstawowe cechy zbiorów (333)
  - Testowanie implementacji zbiorów (336)
- Zbiór w implementacji listowej (342)
- Zbiór haszowany (344)
- Zbiór w implementacji drzewiastej (349)
- Podsumowanie (356)
- Ćwiczenia (356)

### **Rozdział 13. Mapy (357)**

- Koncepcja i zastosowanie map (357)
- Testowanie implementacji map (362)
- Mapa w implementacji listowej (369)
- Mapa w implementacji haszowanej (373)
- Mapa w implementacji drzewiastej (377)
- Podsumowanie (384)
- Ćwiczenia (384)

### **Rozdział 14. Ternarne drzewa wyszukiwawcze (385)**

- Co to jest drzewo ternarne? (385)
  - Wyszukiwanie słowa (386)
  - Wstawianie słowa (389)
  - Poszukiwanie prefiksu (391)
  - Dopasowywanie wzorca (392)
- Drzewa ternarne w praktyce (395)
- Przykład zastosowania - rozwiązywanie krzyżówek (409)
- Podsumowanie (412)
- Ćwiczenie (412)

### **Rozdział 15. B-drzewa (413)**

- Podstawowe własności B-drzew (413)
- Praktyczne wykorzystywanie B-drzew (419)
- Podsumowanie (431)
- Ćwiczenie (431)

### **Rozdział 16. Wyszukiwanie tekstu (433)**

- Interfejs wyszukiwarki łańcuchów (433)
- Zestaw testowy dla wyszukiwarki łańcuchów (435)
- Prymitywny algorytm wyszukiwania (438)
- Algorytm Boyera-Moore'a (441)
  - Tworzenie testów dla algorytmu Boyera-Moore'a (443)
  - Implementowanie algorytmu Boyera-Moore'a (444)
- Iterator dopasowywania wzorca (447)
- Porównanie efektywności wyszukiwania (449)
  - Pomiar efektywności (450)
  - Wyniki eksperymentu (454)
- Podsumowanie (454)

## **Rozdział 17. Dopasowywanie łańcuchów (457)**

- Soundex (457)
- Odległość Levenshteina dwóch słów (468)
- Podsumowanie (477)

## **Rozdział 18. Geometria obliczeniowa (479)**

- Podstawowe pojęcia geometryczne (479)
  - Współrzędne i punkty (479)
  - Linie (481)
  - Trójkąty (481)
  - Znajdowanie punktu przecięcia dwóch linii (482)
- Punkt przecięcia dwóch linii (485)
- Znajdowanie pary najbliższych punktów (499)
- Podsumowanie (510)
- Ćwiczenia (510)

## **Rozdział 19. Optymalizacja pragmatyczna (511)**

- Kiedy optymalizowanie ma sens? (511)
- Profilowanie (513)
- Przykładowy program FileSortingHelper (514)
  - Profilowanie za pomocą modułu hprof (517)
  - Profilowanie za pomocą JMP (520)
- Istota optymalizacji (522)
- Optymalizacja w praktyce (523)
- Podsumowanie (530)

## **Dodatek A Zalecana literatura uzupełniająca (531)**

## **Dodatek B Wybrane zasoby internetowe (533)**

## **Dodatek C Literatura cytowana (535)**

## **Dodatek D Odpowiedzi do ćwiczeń (537)**

## **Skorowidz (585)**