

Opis

Aby projektować aplikacje, nie wystarczy poznać kilka języków programowania i opanować zestaw narzędzi deweloperskich. Konieczna jest jeszcze wiedza o tym, w jaki sposób rozwiązać dany problem, innymi słowy, trzeba poznać algorytmy. Naturalnie w praktyce programistycznej stosuje się ograniczony zestaw algorytmów. Zostały one już dawno przeanalizowane i przetestowane. Opisano je w wielu trudnych książkach najeżonych niezrozumiałymi schematami i dowodami z pogranicza matematyki, statystyki i jeszcze kilku innych nieprzystępnych dziedzin.

Jeśli chcesz po prostu zrozumieć działanie algorytmów, a nie masz ochoty na mozolne przedzieranie się przez setki trudnych stron, to trzymasz w ręku właściwą książkę! Dzięki temu interesującemu, przystępnemu podręcznikowi szybko przyswoisz sobie najważniejsze pojęcia i łatwo zrozumiesz, w jaki sposób algorytmy pomagają w rozwiązywaniu problemów programistycznych. Pokazano w nim słabe i mocne strony najważniejszych algorytmów. Nie zabrakło przydatnych schematów i przykładowych fragmentów kodu napisanego w Pythonie. Książkę tę szczególnie docenią programiści samoucy, inżynierowie i każdy, kto chce zyskać wiedzę o algorytmach.

W tej książce przedstawiono między innymi:

- wyjaśnienie takich pojęć, jak tablice skrótów, listy powiązane, rekurencja,
- algorytmy sortowania, problem komiwojażera, algorytmy zachłanne,
- analiza szybkości algorytmów metodą dużego O ,
- algorytmy grafów, w tym algorytm wyszukiwania wszereż i algorytm Dijkstry,
- algorytm KNN służący do uczenia maszynowego.

Poznaj algorytmy i przekonaj się, jak bardzo są przydatne!

Aditya Bhargava programuje od ponad dwudziestu lat. Jako nastolatek pisał gry wideo w językach Basic i ActionScript. Pracował w kilku startupach. Obecnie jest programistą w [Etsy.com](https://www.etsy.com). Oprócz tego od kilku lat uczy programowania, z powodzeniem przedstawiając trudne koncepcje i idee w taki sposób, aby ich zrozumienie przychodziło bez trudu. Bhargava interesuje się sztuką, literaturą i oczywiście programowaniem.

Spis treści

- Przedmowa (xiii)
- Podziękowania (xiv)
- O książce (xv)

1. Wprowadzenie do algorytmów (1)

- Wprowadzenie (1)
 - Czego nauczysz się o wydajności (2)
 - Czego nauczysz się o rozwiązywaniu problemów (2)

- Wyszukiwanie binarne (3)
 - Lepszy sposób wyszukiwania (5)
 - Czas wykonywania (10)
- Notacja dużego O (10)
 - Czas wykonywania algorytmów rośnie w różnym tempie (11)
 - Wizualizacja różnych czasów wykonywania (13)
 - Notacja dużego O określa czas działania w najgorszym przypadku (15)
 - Kilka typowych czasów wykonywania (15)
 - Problem komiwojażera (17)
- Powtórzenie (19)

2. Sortowanie przez wybieranie (21)

- Jak działa pamięć (22)
- Tablice i listy powiązane (24)
 - Listy powiązane (25)
 - Tablice (26)
 - Terminologia (27)
 - Wstawianie elementów w środku listy (29)
 - Usuwanie elementów (30)
- Sortowanie przez wybieranie (32)
- Powtórzenie (36)

3. Rekurencja (37)

- Rekurencja (38)
- Przypadki podstawowy i rekurencyjny (40)
- Stos (42)
 - Stos wywołań (43)
 - Stos wywołań z rekurencją (45)
- Powtórzenie (50)

4. Szybkie sortowanie (51)

- "Dziel i rządź" (52)
- Sortowanie szybkie (60)
- Jeszcze raz o notacji dużego O (66)
 - Sortowanie przez scalanie a sortowanie szybkie (67)
 - Przypadki średni i najgorszy (68)
- Powtórzenie (72)

5. Tablice skrótów (73)

- Funkcje obliczania skrótów (76)
- Zastosowania tablic skrótów (79)
 - Przeszukiwanie tablic skrótów (80)
 - Zapobieganie powstawaniu duplikatów elementów (81)
 - Tablice skrótów jako pamięć podręczna (83)
 - Powtórzenie wiadomości (86)
- Kolizje (86)

- Wydajność (88)
 - Współczynnik zapelnienia (90)
 - Dobra funkcja obliczania skrótów (92)
- Powtórzenie (94)

6. Przeszukiwanie wszere (95)

- Wprowadzenie do grafów (96)
- Czym jest graf (98)
- Wyszukiwanie wszere (99)
 - Szukanie najkrótszej drogi (102)
 - Kolejki (103)
- Implementacja grafu (105)
- Implementacja algorytmu (107)
 - Czas wykonywania (111)
- Powtórzenie (114)

7. Algorytm Dijkstry (115)

- Posługiwanie się algorytmem Dijkstry (116)
- Terminologia (120)
- Szukanie funduszy na fortepian (122)
- Krawędzie o wadze ujemnej (128)
- Implementacja (131)
- Powtórzenie (140)

8. Algorytmy zachłanne (141)

- Plan zajęć w sali lekcyjnej (142)
- Problem plecaka (144)
- Problem pokrycia zbioru (146)
 - Algorytmy aproksymacyjne (147)
- Problemy NP-zupełne (152)
 - Problem komiwojażera krok po kroku (153)
 - Trzy miasta (154)
 - Cztery miasta (155)
 - Jak rozpoznać, czy problem jest NP-zupełny (158)
- Powtórzenie wiadomości (160)

9. Programowanie dynamiczne (161)

- Problem plecaka (161)
 - Proste rozwiązanie (162)
 - Programowanie dynamiczne (163)
- Pytania dotyczące problemu plecaka (171)
 - Co się dzieje, gdy zostanie dodany element (171)
 - Jaki będzie skutek zmiany kolejności wierszy (174)
 - Czy siatkę można wypełniać wg kolumn zamiast wierszy (174)
 - Co się stanie, gdy doda się mniejszy element (174)
 - Czy można ukraść ułamek przedmiotu (175)

- Optymalizacja planu podróży (175)
- Postępowanie z wzajemnie zależnymi przedmiotami (177)
- Czy możliwe jest, aby rozwiązanie wymagało więcej niż dwóch podplecaków (177)
- Czy najlepsze rozwiązanie zawsze oznacza całkowite wypełnienie plecaka? (178)
- Najdłuższa wspólna część łańcucha (178)
 - Przygotowanie siatki (179)
 - Wypełnianie siatki (180)
 - Najdłuższa wspólna podsekwencja (183)
 - Najdłuższa wspólna podsekwencja - rozwiązanie (184)
- Powtórzenie (186)

10. K najbliższych sąsiadów (187)

- Klasyfikacja pomarańczy i grejpfrutów (187)
- Budowa systemu rekomendacji (189)
 - Wybór cech (191)
 - Regresja (195)
 - Wybieranie odpowiednich cech (198)
- Wprowadzenie do uczenia maszynowego (199)
 - Optyczne rozpoznawanie znaków (199)
 - Budowa filtra spamu (200)
 - Przewidywanie cen akcji (201)
- Powtórzenie (201)

11. Co dalej (203)

- Drzewa (203)
- Odwrócone indeksy (206)
- Transformata Fouriera (207)
- Algorytmy równoległe (208)
- MapReduce (209)
 - Do czego nadają się algorytmy rozproszone (209)
 - Funkcja map (209)
 - Funkcja reduce (210)
- Filtry Blooma i HyperLogLog (211)
 - Filtry Blooma (212)
 - HyperLogLog (213)
- Algorytmy SHA (213)
 - Porównywanie plików (214)
 - Sprawdzanie haseł (215)
- Locality-sensitive hashing (216)
- Wymiana kluczy Diffiego-Hellmana (217)
- Programowanie liniowe (218)
- Epilog (219)

Rozwiązania ćwiczeń (221)

- Skorowidz (235)

