

Spis treści

Przedmowa

Cechy charakterystyczne

Algorytmy

Typy danych

Zastosowania

Podejście naukowe

Szeroki zakres

Witryna poświęcona książce

Elektroniczne streszczenie

Pełne implementacje

Ćwiczenia i odpowiedzi

Dynamiczne wizualizacje

Materiały do kursu

Odnośniki do powiązanych materiałów

Wykorzystanie w programie nauczania

Kontekst

Podziękowania

Rozdział 1. Podstawy

Algorytmy

Podsumowanie zagadnień

Podstawy

Sortowanie

Wyszukiwanie

Grafy

Łańcuchy znaków

Kontekst

1.1. Podstawowy model programowania

Podstawowa struktura programu Javy

Proste typy danych i wyrażenia

Wyrażenia

Konwersja typów

Porównania

Inne typy proste

Instrukcje

Deklaracje

Przypisania

Instrukcje warunkowe

Pętle

Instrukcje break i continue

Zapis skrócony

Deklaracje inicjujące

Przypisania niejawne

Bloki z jedną instrukcją

Notacja for

Tablice

Tworzenie i inicjowanie tablic

Krótki zapis

Używanie tablicy

Utożsamianie nazw (ang. aliasing)

Tablice dwuwymiarowe

Metody statyczne

- Definiowanie metody statycznej
- Wywoływanie metod statycznych
- Cechy metod
- Rekurencja
- Podstawowy model programowania
- Programowanie modułowe
- Testy jednostkowe
- Biblioteki zewnętrzne

Interfejsy API

- Przykład
- Biblioteki Javy
- Opracowane przez nas biblioteki standardowe
- Własne biblioteki

Łańcuchy znaków

- Złączanie
- Konwersja
- Konwersja automatyczna
- Argumenty wiersza poleceń

Wejście i wyjście

- Polecenia i argumenty
- Standardowe wyjście
- Sformatowane dane wyjściowe
- Standardowe wejście
- Przekierowywanie i potoki
- Dane wejściowe i wyjściowe z pliku
- Standardowe rysowanie (podstawowe metody)
- Standardowe rysowanie (metody pomocnicze)

Wyszukiwanie binarne

- Wyszukiwanie binarne
- Klient wspomagający tworzenie aplikacji
 - Wyszukiwanie binarne
- Stosowanie białych list
- Wydajność

Perspektywa

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

1.2. Abstrakcja danych

Korzystanie z abstrakcyjnych typów danych

- Interfejs API abstrakcyjnego typu danych
- Metody odziedziczone
- Kod klienta
- Obiekty
- Tworzenie obiektów
- Wywoływanie metod egzemplarza
- Korzystanie z obiektów
- Instrukcje przypisania
- Obiekty jako argumenty
- Obiekty jako zwracane wartości
- Tablice to obiekty

- Tablice obiektów
- Przykładowe abstrakcyjne typy danych
 - Obiekty geometryczne
 - Przetwarzanie informacji
 - Łańcuchy znaków
 - Ponownie o wejściu i wyjściu
- Implementowanie abstrakcyjnych typów danych
 - Zmienne egzemplarza
 - Konstruktory
 - Metody egzemplarza
 - Zasięg
 - Interfejs API, klienty i implementacje
- Więcej implementacji typów ADT
 - Date
 - Utrzymywanie wielu implementacji
 - Akumulator
 - Wizualny akumulator
- Projektowanie typu danych
 - Hermetyzacja
 - Projektowanie interfejsów API
 - Algorytmy i abstrakcyjne typy danych
 - Dziedziczenie interfejsu
 - Dziedziczenie implementacji
 - Przekształcanie łańcuchów znaków
 - Typy nakładkowe
 - Równość
 - Zarządzanie pamięcią
 - Niezmiennność
 - Projektowanie kontraktowe
 - Wyjątki i błędy
 - Asercje
 - Podsumowanie
- Pytania i odpowiedzi
- Ćwiczenia
- Problemy do rozwiązania
- 1.3. Wielozbiory, kolejki i stosy
 - Interfejsy API
 - Typy generyczne
 - Autoboxing
 - Kolekcje z możliwością iterowania
 - Wielozbiory
 - Kolejki FIFO
 - Stosy
 - Przetwarzanie wyrażeń arytmetycznych
- Implementowanie kolekcji
 - Stos o stałej pojemności
 - Typy generyczne
 - Zmiana wielkości tablicy
 - Zbędne referencje
 - Iterowanie
- Listy powiązane
 - Rekord węzła

- Budowanie listy powiązanej
- Wstawianie na początek
- Usuwanie z początku
- Wstawianie na koniec
- Wstawianie i usuwanie na innych pozycjach
- Przechodzenie
- Implementacja stosu
- Implementacja kolejki
- Implementacja wielozbiorów
- Przegląd
 - Struktury danych
- Pytania i odpowiedzi
- Ćwiczenia
- Ćwiczenia dotyczące list powiązanych
- Problemy do rozwiązania
- 1.4. Analizy algorytmów
 - Metoda naukowa
 - Obserwacje
 - Przykład
 - Stoper
 - Analizy danych eksperymentalnych
- Modele matematyczne
 - Przybliżenia z tyldą
 - Przybliżony czas wykonania
 - Hipotezy dotyczące tempa wzrostu
 - Analizy algorytmów
 - Model kosztów
 - Podsumowanie
- Kategorie tempa wzrostu
 - Stałe
 - Logarytmiczne
 - Liniowe
 - Liniowo-logarytmiczne
 - Kwadratowe
 - Sześciennie
 - Wykładnicze
- Projektowanie szybszych algorytmów
 - Rozgrzewka: sumy par
 - Szybki algorytm dla sum trójek
 - Dolne ograniczenia
- Eksperymenty ze stosunkiem czasu wykonania dla podwojonych danych
 - Szacowanie możliwości rozwiązania dużych problemów
 - Szacowanie korzyści z zastosowania szybszego komputera
- Zastrzeżenia
 - Duże stałe
 - Pętla wewnętrzna, która nie dominuje
 - Czas wykonania instrukcji
 - Uwzględnianie systemu
 - Zbyt małe różnice
 - Duża zależność od danych wejściowych
 - Problemy o wielu parametrach
- Radzenie sobie z zależnością od danych wejściowych

- Modele danych wejściowych
- Gwarancje wydajności dla najgorszego przypadku
- Algorytmy z randomizacją
- Ciągi operacji
- Analizy z uwzględnieniem amortyzacji

Pamięć

- Obiekty
- Listy powiązane
- Tablice
- Obiekty typu String
- Wartości typu String i podłańcuchy

Perspektywa

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

1.5. Studium przypadku problem union-find

Dynamiczne określanie połączeń

Sieci

Równoznaczność nazw zmiennych

Zbiory matematyczne

Implementacje

Szybka metoda find

Analizy techniki z szybką metodą find

Technika z szybką metodą union

Reprezentacja lasu drzew

Analiza techniki z szybką metodą union()

Szybka metoda union() z wagami

Analiza szybkiej metody union() z wagami

Optymalne algorytmy

Wykresy kosztów z amortyzacją

Perspektywy

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

Rozdział 2. Sortowanie

2.1. Podstawowe metody sortowania

Reguły

Sprawdzanie poprawności

Czas wykonania

Dodatkowa pamięć

Typy danych

Sortowanie przez wybieranie

Czas wykonania jest niezależny od danych wejściowych

Potrzebna jest minimalna liczba przestawień

Sortowanie przez wstawianie

Wizualizacja działania algorytmów sortujących

Porównywanie dwóch algorytmów sortujących

Sortowanie Shella

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

2.2. Sortowanie przez scalanie

Abstrakcyjne scalanie w miejscu

Zstępujące sortowanie przez scalanie

Stosowanie sortowania przez wstawianie dla małych podtablic

Sprawdzanie, czy tablica jest już uporządkowana

Eliminowanie kopiowania danych do tablicy pomocniczej

Wstępujące sortowanie przez scalanie

Złożoność sortowania

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

2.3. Sortowanie szybkie

Podstawowy algorytm

Podział w miejscu

Pozostawanie w granicach

Zachowanie losowości

Kończenie pracy pętli

Elementy z kluczami równymi kluczowi elementu osiowego

Kończenie rekurencji

Cechy związane z wydajnością

Usprawnienia algorytmu

Przełączanie na sortowanie przez wstawianie

Podział w miejscu mediany trzech elementów

Sortowanie optymalne ze względu na entropię

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

2.4. Kolejki priorytetowe

Interfejs API

Klient kolejki priorytetowej

Podstawowe implementacje

Reprezentacja w postaci nieuporządkowanej tablicy

Reprezentacja w postaci uporządkowanej tablicy

Reprezentacje w postaci listy powiązanej

Definicje kopca

Reprezentacja sterty binarnej

Algorytmy oparte na kopcach

Przywracanie struktury kopca przy przechodzeniu do góry (wypływanie)

Przywracanie struktury kopca przy przechodzeniu w dół (zatapianie)

Kopce a-arne

Zmiana wielkości tablicy

Niezmiennność kluczy

Indeksowana kolejka priorytetowa

Klient indeksowanej kolejki priorytetowej

Sortowanie przez kopcowanie

Tworzenie kopca

Sortowanie

Zatapianie do poziomu dna i późniejsze wypływanie

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

2.5. Zastosowania

Sortowanie różnych typów danych

Przykład transakcje

Sortowanie wskaźników

Niezmiennie klucze

Niekosztowne przestawienia

Różne porządki

Elementy o wielu kluczach

Kolejki priorytetowe z komparatorami

Stabilność

Który algorytm sortowania mam zastosować?

Sortowanie typów prostych

Sortowanie systemowe Javy

Redukcje

Powtórzenia

Permutacje

Redukcje oparte na kolejkach priorytetowych

Mediana i inne miary statystyczne

Krótki przegląd zastosowań sortowania

Przetwarzanie komercyjne

Wyszukiwanie informacji

Badania operacyjne

Symulacje oparte na zdarzeniach

Obliczenia numeryczne

Wyszukiwanie kombinatoryczne

Algorytmy Prima i Dijkstry

Algorytm Kruskala

Kompresja Huffmana

Algorytmy przetwarzania łańcuchów znaków

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

Rozdział 3. Wyszukiwanie

3.1. Tablice symboli

Interfejs API

Typy generyczne

Powtarzające się klucze

Klucze o wartości null

Wartości null

Usuwanie

Metody skrócone

Iteracja

Równość kluczy

Uporządkowane tablice symboli

Minimum i maksimum

Podłoga i sufit

Pozycja i wybieranie

- Zapytania zakresowe
- Wyjątkowe przypadki
- Metody skrócone
- Równość kluczy (raz jeszcze)
- Model kosztów
- Przykładowe klienty
 - Klient testowy
 - Klient do pomiaru wydajności
- Sekwencyjne przeszukiwanie nieuporządkowanych list powiązanych
- Wyszukiwanie binarne w uporządkowanej tablicy
 - Wyszukiwanie binarne
 - Inne operacje
- Analizy wyszukiwania binarnego
- Przegląd wstępny
- Pytania i odpowiedzi
- Ćwiczenia
- Problemy do rozwiązania
- Eksperymenty
- 3.2. Drzewa wyszukiwań binarnych
 - Podstawowa implementacja
 - Reprezentacja
 - Wyszukiwanie
 - Wstawianie
 - Rekurencja
 - Analizy
 - Eksperymenty
 - Metody oparte na uporządkowaniu i usuwanie
 - Minimum i maksimum
 - Podłoga i sufit
 - Wybieranie
 - Pozycja
 - Usuwanie minimum i maksimum
 - Usuwanie
 - Zapytania zakresowe
 - Analiza
 - Pytania i odpowiedzi
 - Ćwiczenia
 - Problemy do rozwiązania
 - Eksperymenty
- 3.3. Zbalansowane drzewa wyszukiwań
 - Drzewa wyszukiwań 2-3
 - Wyszukiwanie
 - Wstawianie do węzła podwójnego
 - Wstawianie do drzewa składającego się z jednego węzła potrójnego
 - Wstawianie do węzła potrójnego, którego rodzicem jest węzeł podwójny
 - Wstawianie do węzła potrójnego, którego rodzicem jest węzeł potrójny
 - Podział korzenia
 - Transformacje lokalne
 - Właściwości globalne
 - Czerwono-czarne drzewa BST
 - Zapisywanie węzłów potrójnych
 - Równoważna definicja

- Zależność 1 do 1
- Reprezentacja kolorów
- Rotacje
- Ponowne ustawianie odnośnika w rodzicu po rotacji
- Wstawianie do jednego węzła podwójnego
- Wstawianie do węzła podwójnego w dolnej części drzewa
- Wstawianie do drzewa o trzech kluczach (do węzła potrójnego)
- Zachowanie czarnego koloru korzenia
- Wstawianie do węzła potrójnego na dole drzewa
- Przenoszenie czerwonego odnośnika w górę drzewa
- Implementacja
- Usuwanie
 - Zstępujące drzewa 2-3-4
 - Usuwanie minimum
 - Usuwanie
- Cechy czerwono-czarnych drzew BST
 - Analizy
 - Interfejs API dla uporządkowanej tablicy symboli
- Pytania i odpowiedzi
- Ćwiczenia
- Problemy do rozwiązania
- Eksperymenty
- 3.4. Tablice z haszowaniem
 - Funkcje haszujące
 - Typowy przykład
 - Dodatnie liczby całkowite
 - Liczby zmiennoprzecinkowe
 - Łańcuchy znaków
 - Klucze złożone
 - Konwencje stosowane w Javie
 - Przekształcanie wartości funkcji hashCode() na indeks tablicy
 - Metoda hashCode() definiowana przez użytkownika
 - Programowa pamięć podręczna
 - Haszowanie metodą łańcuchową
 - Wielkość tablicy
 - Usuwanie
 - Operacje na kluczach uporządkowanych
 - Haszowanie z wykorzystaniem próbkowania liniowego
 - Usuwanie
 - Grupowanie
 - Analiza próbkowania liniowego
 - Zmienianie wielkości tablicy
 - Metoda łańcuchowa
 - Analizy z uwzględnieniem amortyzacji
 - Pamięć
 - Pytania i odpowiedzi
 - Ćwiczenia
 - Problemy do rozwiązania
 - Eksperymenty
- 3.5. Zastosowania
 - Którą implementację tablicy symboli powinienem zastosować?
 - Typy proste

- Powtarzające się klucze
- Biblioteki Javy
- Interfejs API dla zbiorów
- Usuwanie powtórzeń
- Białe i czarne listy
- Klienty używające słownika
- Klienty używające indeksu
- Indeks odwrotny
- Wektory rzadkie
- Pytania i odpowiedzi
- Ćwiczenia
- Problemy do rozwiązania
- Eksperymenty
- Rozdział 4. Grafy
 - Mapy
 - Zawartość stron WWW
 - Obwody
 - Harmonogramy
 - Handel
 - Dopasowywanie
 - Sieci komputerowe
 - Oprogramowanie
 - Sieci społecznościowe
 - 4.1. Grafy nieskierowane
 - Anomalie
 - Słowniczek
 - Typ danych dla grafów nieskierowanych
 - Możliwe reprezentacje
 - Listy sąsiedztwa
 - Wzorce projektowe z zakresu przetwarzania grafów
 - Przeszukiwanie w głąb
 - Przeszukiwanie labiryntu
 - Rozgrzewka
 - Alejkę jednokierunkowe
 - Śledzenie działania metody DFS
 - Szczegółowy ślad przeszukiwania w głąb
 - Wyznaczanie ścieżek
 - Implementacja
 - Szczegółowy ślad
 - Przeszukiwanie wszerz
 - Implementacja
 - Spójne składowe
 - Implementacja
 - Algorytmy Union-Find
 - Grafy symboli
 - Interfejs API
 - Klient testowy
 - Implementacja
 - Stopnie oddalenia
 - Podsumowanie
 - Pytania i odpowiedzi
 - Ćwiczenia

Problemy do rozwiązania

Eksperymenty

4.2. Grafy skierowane

Słownictwo

Typ danych Digraph

Reprezentacja

Format danych wejściowych

Odwracanie digrafu

Nazwy symboliczne

Osiągalność w digrafach

Przywracanie pamięci metodą znac i zmiataj (ang. mark and sweep)

Znajdowanie ścieżek w grafach

Cykle i grafy DAG

Problem szeregowania zadań

Cykle w digrafach

Kolejność przy przeszukiwaniu w głąb i sortowanie topologiczne

Silna spójność w digrafach

Silnie spójne składowe

Przykładowe zastosowania

Algorytm Kosaraju

Osiągalność po raz wtóry

Podsumowanie

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

4.3. Minimalne drzewa rozpinające

Założenia

Przestrzegane zasady

Właściwość przekroju

Algorytm zachłanny

Typ danych dla grafów ważonych

Porównywanie krawędzi według wag

Krawędzie równoległe

Pętle własne

Interfejs API do wyznaczania drzew MST i klient testowy

Klient testowy

Dane testowe

Algorytm Prima

Struktury danych

Tworzenie zbioru krawędzi przekroju

Implementacja

Czas wykonania

Zachłanna wersja algorytmu Prima

Algorytm Kruskala

Perspektywa

Uwagi historyczne

Algorytm działający w czasie liniowym

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

4.4. Najkrótsze ścieżki

- Cechy najkrótszych ścieżek

 - Drzewo najkrótszych ścieżek

- Typy danych dla digrafów ważonych

 - Interfejs API do wyznaczania najkrótszych ścieżek

 - Klient testowy

 - Struktury danych do wyznaczania najkrótszych ścieżek

 - Relaksacja krawędzi

 - Relaksacja wierzchołka

 - Metody obsługi zapytań od klientów

- Teoretyczne podstawy algorytmów wyznaczania najkrótszych ścieżek

 - Warunki optymalności

 - Sprawdzanie

 - Ogólny algorytm

- Algorytm Dijkstry

 - Struktury danych

 - Inna perspektywa

 - Odmiany

- Acykliczne digrafy ważne

 - Najdłuższe ścieżki

 - Szeregowanie równoległych zadań

 - Szeregowanie zadań równoległych z uwzględnieniem względnych terminów granicznych

- Najkrótsze ścieżki w ogólnych digrafach ważonych

 - Próba numer I

 - Próba numer II

 - Cykle ujemne

 - Próba numer III

 - Algorytm Bellmana-Forda oparty na kolejce

 - Implementacja

 - Wagi ujemne

 - Wykrywanie cykli ujemnych

 - Arbitraż

- Perspektywa

 - Uwagi historyczne

- Pytania i odpowiedzi

- Ćwiczenia

- Problemy do rozwiązania

- Eksperymenty

Rozdział 5. Łańcuchy znaków

- Przetwarzanie informacji

- Badania nad genomem

- Systemy komunikacji

- Systemy programowania

- Zasady gry

 - Znaki

 - Niezmiennność

 - Indeksowanie

 - Długość

 - Podłańcuch

 - Złączanie

 - Tablice znaków

- Alfabet

Tablice indeksowane znakami

Liczby

5.1. Sortowanie łańcuchów znaków

Sortowanie przez zliczanie

Zliczanie wystąpień

Przekształcanie liczb wystąpień na indeksy

Rozdzielanie danych

Kopiowanie z powrotem

Sortowanie łańcuchów znaków metodą LSD

Sortowanie łańcuchów znaków metodą MSD

Konwencja wykrywania koń--ca łańcucha znaków

Określony alfabet

Małe podtablice

Równe klucze

Dodatkowa pamięć

Model losowych łańcuchów znaków

Wydajność

Szybkie sortowanie łańcuchów znaków z podziałem na trzy części

Krótkie podtablice

Ograniczony alfabet

Randomizacja

Wydajność

Przykład dzienniki sieciowe

Z którego algorytmu sortowania łańcuchów znaków powinienem korzystać?

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

5.2. Drzewa trie

Drzewa trie

Podstawowe cechy

Przeszukiwanie drzewa trie

Wstawianie do drzewa trie

Reprezentacja węzłów

Określanie wielkości

Pobieranie kluczy

Dopasowywanie symboli wieloznacznych

Najdłuższy przedrostek

Usuwanie

Alfabet

Cechy drzew trie

Ograniczenia czasowe dla najgorszego przypadku przy wyszukiwaniu i wstawianiu

Ograniczenia oczekiwanego czasu nieudanego wyszukiwania

Pamięć

Jednokierunkowe gałęzie

Trójkowe drzewa wyszukiwań (drzewa TST)

Wyszukiwanie i wstawianie

Cechy drzew TST

Pamięć

Koszt wyszukiwania

Alfabet

Dopasowywanie przedrostków, pobieranie kluczy i dopasowywanie do symboli wieloznacznych

Usuwanie

Hybrydowe drzewa TST

Jednokierunkowe gałęzie

Której implementacji tablicy symboli z łańcuchami znaków powinienem używać?

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

5.3. Wyszukiwanie podłańcuchów

Krótką historią

Wyszukiwanie podłańcuchów metodą ataku siłowego

Wyszukiwanie podłańcuchów metodą Knutha-Morrisa-Pratta

Cofanie wskaźnika wzorca

Wyszukiwanie metodą KMP

Symulacja deterministycznego automatu skończonego

Tworzenie automatu DFA

Wyszukiwanie podłańcuchów metodą Boyera-Moore'a

Heurystyka obsługi niedopasowania znaku

Punkt wyjścia

Wyszukiwanie podłańcuchów

Wyszukiwanie metodą odcisków palców (metoda Rabina-Karpa)

Podstawowy plan

Obliczanie wartości funkcji haszującej

Kluczowy pomysł

Implementacja

Sztuczka poprawności metody Monte Carlo

Podsumowanie

Pytania i odpowiedzi

Ćwiczenia

Problemy do rozwiązania

Eksperymenty

5.4. Wyrażenia regularne

Opisywanie wzorców za pomocą wyrażeń regularnych

Złączanie (konkatenacja)

Lub

Domknięcie

Nawiasy

Skróty

Deskryptory zbiorów znaków

Skróty dla domknięcia

Sekwencje ucieczki

Zastosowania wyrażeń regularnych

Wyszukiwanie podłańcuchów

Sprawdzanie poprawności

Narzędzia programisty

Badania nad genomem

Wyszukiwanie

Możliwości

Ograniczenia

Niedeterministyczne automaty skończone

- Symulowanie działania automatu NFA
 - Reprezentacja
 - Symulowanie działania automatu NFA i osiągalność
- Tworzenie automatu NFA odpowiadającego wyrażeniu regularnemu
 - Złączanie
 - Nawiasy
 - Domknięcie
 - Wyrażenie z lub
- Pytania i odpowiedzi
- Ćwiczenia
- Problemy do rozwiązania
- 5.5. Kompresja danych
 - Reguły działania
 - Podstawowy model
 - Odczyt i zapis danych binarnych
 - Binarne wejście i wyjście
 - Przykład
 - Zrzuty binarne
 - Kodowanie ASCII
 - Ograniczenia
 - Uniwersalne algorytmy kompresji danych
 - Nierozstrzygalność
 - Rozgrzewka genom
 - Dane o genomie
 - Kompresja za pomocą kodu 2-bitowego
 - Rozpakowywanie dla kodu 2-bitowego
 - Kodowanie długości serii
 - Bitmapy
 - Implementacja
 - Zwiększanie rozdzielczości bitmap
 - Kompresja Huffmana
 - Kody bezprefiksowe o zmiennej długości
 - Reprezentacja kodów bezprefiksowych za pomocą drzewa trie
 - Ogólne omówienie
 - Węzły drzewa trie
 - Rozpakowywanie za pomocą kodów bezprefiksowych
 - Kompresja za pomocą kodów bezprefiksowych
 - Tworzenie drzewa trie
 - Optymalność
 - Zapis i odczyt drzewa trie
 - Implementacja kompresji Huffmana
 - Kompresja LZW
 - Przykładowa kompresja LZW
 - Reprezentacja kompresji LZW za pomocą drzewa trie
 - Rozpakowywanie w metodzie LZW
 - Skomplikowana sytuacja
 - Implementacja
 - Pytania i odpowiedzi
 - Ćwiczenia
 - Problemy do rozwiązania
- Rozdział 6. Kontekst
 - Zastosowania komercyjne

- Obliczenia naukowe
- Inżynieria
- Badania operacyjne
- Symulacja sterowana zdarzeniami
 - Model oparty na twardych dyskach
 - Symulacje sterowane czasem
 - Symulacja sterowana zdarzeniami
 - Prognozowanie zdarzeń
 - Efekt zderzenia
 - Unieważnione zdarzenia
 - Cząsteczki
 - Zdarzenia
 - Kod do symulowania ruchu
 - Wydajność
- Drzewa zbalansowane
 - Model kosztów
 - Drzewa zbalansowane (b-drzewa)
 - Konwencje
 - Wyszukiwanie i wstawianie
 - Reprezentacja
 - Wydajność
 - Pamięć
- Tablice przyrostkowe
 - Najdłuższy powtarzający się łańcuch znaków
 - Rozwiązanie oparte na ataku siłowym
 - Rozwiązanie oparte na sortowaniu przyrostków
 - Indeksowanie łańcucha znaków
 - Interfejs API i kod kliencki
 - Implementacja
 - Wydajność
 - Usprawnione implementacje
- Algorytmy dla sieci przepływowych
 - Model fizyczny
 - Definicje
 - Interfejsy API
 - Algorytm Forda-Fulkersona
- Twierdzenie przepływu maksymalnego i przekroju minimalnego
 - Sieć rezydualna
 - Metoda najkrótszej ścieżki powiększającej
 - Wydajność
 - Inne implementacje
- Redukcja
 - Redukcje w obszarze sortowania
 - Redukcje do problemu wyznaczania najkrótszych ścieżek
 - Redukcje do problemu wyznaczania przepływu maksymalnego
 - Programowanie liniowe
- Nierozwiązywalność
 - Podstawowe prace
 - Czas wykonania rosnący wykładniczo
 - Problemy przeszukiwania
- Wybrane problemy przeszukiwania
 - Inne rodzaje problemów

- Łatwe problemy przeszukiwania
- Niedeterminizm
- Podstawowe pytanie
- Redukcje wielomianowe
- NP-zupełność
- Twierdzenie Cooka-Levina
- Klasyfikowanie problemów
- Wybrane znane problemy NP-zupełne
 - Radzenie sobie z NP-zupełnością
 - Ćwiczenia dotyczące symulowania zderzeń
 - Ćwiczenia dotyczące drzew zbalansowanych
 - Ćwiczenia dotyczące tablicy przyrostkowej
 - Ćwiczenia dotyczące przepływu maksymalnego
 - Ćwiczenia dotyczące redukcji i nierozwiązywalności
- Algorytmy
- Klienty