

Przedmowa

1. Zanim wystartujemy

- 1.1 Jak to wcześniej bywało, czyli
- 1.2 Jak to się niedawno odbyło, czyli
- 1.3 Proces koncepcji programów
- 1.4 Poziomy abstrakcji opisu i wybór języka
- 1.5 Poprawność algorytmów

2. Rekurencja

- 2.1 Definicja rekurencji
- 2.2 Ilustracja pojęcia rekurencji
- 2.3 Jak wykonują się programy rekurencyjne?
- 2.4 Niebezpieczeństwa rekurencji
 - 2.4.1 Ciąg Fibonacciego
 - 2.4.2 Stack overflow!
- 2.5 Pułapek ciąg dalszy
 - 2.5.1 Stąd do wieczności
 - 2.5.2 Definicja poprawna, ale...
- 2.6 Typy programów rekurencyjnych
- 2.7. Myślenie rekurencyjne
 - 2.7.1 Spirala
 - 2.7.2 Kwadraty "parzyste"
- 2.8 Uwagi praktyczne na temat technik rekurencyjnych
- 2.9 Zadania
- 2.10 Rozwiązania i wskazówki do zadań

3. Analiza sprawności algorytmów

- 3.1 Dobre samopoczucie użytkownika programu
- 3.2 Przykład 1: Jeszcze raz funkcja silnia
- 3.3 Przykład 2: Zerowanie fragmentu tablicy
- 3.4 Przykład 3: Wpadamy w pułapkę
- 3.5 Przykład 4: Różne typy złożoności obliczeniowej
- 3.6 Nowe zadanie: uprościć obliczenia!
- 3.7 Analiza programów rekurencyjnych
 - 3.7.1 Terminologia
 - 3.7.2 Ilustracja metody na przykładzie
 - 3.7.3 Rozkład "logarytmiczny"
 - 3.7.4 Zmiana dziedziny równania rekurencyjnego
 - 3.7.5 Funkcja Ackermana, czyli coś dla smakoszy
- 3.8 Zadania
- 3.9 Rozwiązania i wskazówki do zadań

4. Algorytmy sortowania

- 4.1 Sortowanie przez wstawianie, algorytm klasy $O(N^2)$
- 4.2 Sortowanie bąbelkowe, algorytm klasy $O(N^2)$
- 4.3 Quicksort, algorytm klasy $O(N \log^2 N)$

- 4.4 Uwagi praktyczne

5. Struktury danych

- 5.1 Listy jednokierunkowe
 - 5.1.1 Realizacja strukturalnych listy jednokierunkowej
 - 5.1.2 Tworzenie listy jednokierunkowej
 - 5.1.3. Listy jednokierunkowe - teoria i rzeczywistość
- 5.2 Tablicowa implementacja list
 - 5.2.1 Klasyczna reprezentacja tablicowa
 - 5.2.2 Metoda tablic równoległych
 - 5.2.3 Listy innych typów
- 5.3 Stos
 - 5.3.1 Zasada działania stosu
- 5.4 Kolejki FIFO
- 5.5 Sterty i kolejki priorytetowe
- 5.6 Drzewa i ich reprezentacje
 - 5.6.1 Drzewa binarne i wyrażenia arytmetyczne
- 5.7 Uniwersalna struktura słownikowa
- 5.8 Zadania
- 5.9 Rozwiązania i wskazówki do zadań

6. Derekursywacja

- 6.1 Jak pracuje kompilator?
- 6.2 Odrobina formalizmu ... nie zaszkodzi!
- 6.3 Kilka przykładów derekursywacji algorytmów
- 6.4 Derekursywacja z wykorzystaniem stosu
 - 6.4.1 Eliminacja zmiennych lokalnych
- 6.5 Metoda funkcji przeciwnych
- 6.6 Klasyczne schematy derekursywacji
 - 6.6.1 Schemat typu while
 - 6.6.2 Schemat typu if ... else
 - 6.6.3 Schemat z podwójnym wywołaniem rekurencyjnym
- 6.7 Podsumowanie

7. Algorytmy przeszukiwania

- 7.1 Przeszukiwanie liniowe
- 7.2 Przeszukiwanie binarne
- 7.3 Transformacja kluczowa
 - 7.3.1 W poszukiwaniu funkcji H
 - 7.3.2 Najbardziej znane funkcje H
 - 7.3.3 Obsługa konfliktów dostępu
 - 7.3.4 Zastosowania transformacji kluczowej
 - 7.3.5 Podsumowanie metod transformacji kluczowej

8. Przeszukiwanie tekstów

- 8.1 Algorytm typu brute-force

- 8.2 Nowe algorytmy poszukiwań
 - 8.2.1 Algorytm K-M-P
 - 8.2.2 Algorytm Boyera i Moore'a
 - 8.2.3 Algorytm Rabina i Karpa

9. Zaawansowane techniki programowania

- 9.1 Program typu "dziel-i-rządź"
 - 9.1.1 Odszukiwanie minimum i maksimum w tablicy liczb
 - 9.1.2 Mnożenie macierzy o rozmiarze $N \times N$
 - 9.1.3 Mnożenie liczb całkowitych
 - 9.1.4 Inne znane algorytmy "dziel-i-rządź"
- 9.2 Algorytmy "żarłoczne", czyli
 - 9.2.1 Problem plecakowy, czyli niełatwe jest życie turysty-piechura
- 9.3 Programowanie dynamiczne
- 9.4 Uwagi bibliograficzne

10. Elementy algorytmiki grafów

- 10.1 Kilka definicji i pojęć na temat grafów
- 10.2 Sposoby reprezentacji grafów
- 10.3 Podstawowe operacje na grafach
- 10.4 Algorytm Roy-Warshalla
- 10.5 Algorytm Floyda
- 10.6 Podsumowanie

11 Zadania różne

- 11.1 Teksty zadań
- 11.2 Rozwiązania

Dodatek A. Poznaj C++ w pięć minut!

Spis ilustracji

Spis tablic

Literatura