
Spis treści

Przedmowa	7
Wprowadzenie	9
1. Rozwiązywanie problemów	13
Czym jest algorytm?	13
Znajdowanie największej wartości w dowolnej liście	16
Zliczanie kluczowych operacji	17
Modele pozwalają prognozować wydajność algorytmu	18
Znajdowanie dwóch największych wartości na dowolnej liście	23
Algorytm pucharowy	26
Złożoność czasowa i pamięciowa	32
Podsumowanie	33
Ćwiczenia	34
2. Analiza algorytmów	37
Używanie modeli empirycznych do prognozowania wydajności	38
Mnożenie można wykonywać szybciej	40
Klasy złożoności	41
Analiza asymptotyczna	43
Zliczanie wszystkich operacji	46
Zliczanie wszystkich bajtów	47
Gdy zamykają się jedne drzwi, otwierają się inne	48
Wyszukiwanie binarne w tablicy	49
Prawie tak łatwe jak π	50
Dwie pieczenie na jednym ogniu	51
Łączenie wszystkich elementów	55
Dopasowywanie do krzywej a dolna i górna granica	57
Podsumowanie	58
Ćwiczenia	58

3. Lepsze życie dzięki lepszemu haszowaniu	61
Łączenie wartości z kluczami	61
Funkcje haszujące i skróty	65
Tablica z haszowaniem dla par (klucz, wartość)	67
Wykrywanie i rozwiązywanie kolizji za pomocą próbkowania liniowego	68
Tworzenie odrębnych łańcuchów dzięki listom powiązanym	73
Usuwanie elementu z listy powiązanej	77
Ocena wydajności	77
Zwiększanie rozmiaru tablic z haszowaniem	80
Analiza wydajności dynamicznych tablic z haszowaniem	84
Haszowanie doskonałe	86
Iteracyjne pobieranie par (klucz, wartość)	88
Podsumowanie	90
Ćwiczenia	90
4. Wędrówka po kopcu	95
Kopce binarne typu max	101
Wstawianie elementu (wartość, priorytet)	104
Usuwanie wartości o najwyższym priorytecie	106
Reprezentowanie kopca binarnego za pomocą tablicy	109
Implementacja „wypływania” i „zatapiańia”	110
Podsumowanie	114
Ćwiczenia	115
5. Sortowanie bez tajemnic	117
Sortowanie przez przestawianie	118
Sortowanie przez wybieranie	119
Budowa algorytmu sortowania o złożoności kwadratowej	121
Analizowanie wydajności sortowania przez wstawianie i sortowania przez wybieranie	123
Rekurencja oraz podejście dziel i rządź	124
Sortowanie przez scalanie	129
Sortowanie szybkie	132
Sortowanie przez kopcowanie	135
Porównanie wydajności algorytmów o złożoności $O(N \log N)$	138
Algorytm timsort	139
Podsumowanie	141
Ćwiczenie	141

6. Drzewa binarne — nieskończoność na wyciągnięcie ręki	143
Wprowadzenie	144
Binarne drzewa poszukiwań	148
Szukanie wartości w binarnym drzewie poszukiwań	153
Usuwanie wartości z binarnego drzewa poszukiwań	154
Przechodzenie drzewa binarnego	157
Analiza wydajności binarnych drzew poszukiwań	159
Samoorganizujące się drzewa binarne	161
Analiza wydajności drzew samoorganizujących się	168
Używanie drzewa binarnego jako tablicy symboli (klucz, wartość)	169
Używanie drzewa binarnego jako kolejki priorytetowej	170
Podsumowanie	173
Ćwiczenia	174
7. Grafy — połącz punkty	177
Grafy służą do wydajnego zapisywania przydatnych informacji	177
Znajdowanie drogi w labiryncie za pomocą przeszukiwania w głąb	181
Inna strategia — przeszukiwanie wszerek	187
Grafy skierowane	193
Grafy z wagami krawędzi	199
Algorytm Dijkstry	202
Najkrótsze ścieżki dla wszystkich par	211
Algorytm Floyd-Warshalla	214
Podsumowanie	218
Ćwiczenia	218
8. Podsumowanie	221
Wbudowane typy Pythona	222
Implementowanie stosu w Pythonie	224
Implementowanie kolejek w Pythonie	225
Implementacje kopca i kolejki priorytetowej	226
Dalsza nauka	227