

Spis treści

PRZEDMOWA	15
PODZIĘKOWANIA	17
WPROWADZENIE	19
1	
TABLICE MIESZAJĄCE	33
Problem 1.: Płatki śniegu	34
Problem	34
Uproszczenie problemu	36
Rozwiązywanie podstawowego problemu	38
Rozwiązanie 1.: Porównywanie parami	41
Rozwiązanie 2.: Zmniejszenie liczby wykonywanych operacji	46
Tablice mieszające	52
Projekt tablicy mieszającej	52
Dlaczego warto używać tablic mieszających?	55
Problem 2.: Chaos w hasłach	55
Problem	56
Rozwiązanie 1.: Sprawdzanie wszystkich haseł	57
Rozwiązanie 2.: Użycie tablicy mieszającej	58
Problem 3.: Sprawdzanie pisowni — usuwanie litery	64
Problem	64
Rozważania o zastosowaniu tablic mieszających	65
Rozwiązanie doraźne	67
Podsumowanie	71
Uwagi	71

DRZEWA I REKURENCJA	73
Problem 1.: Halloweenowy łup	74
Problem	74
Drzewa binarne	75
Rozwiązywanie problemu dla przykładowego drzewa	77
Reprezentacja drzew binarnych	78
Zbieranie wszystkich cukierków	83
Zupełnie inne rozwiązanie	89
Przechodzenie minimalnej liczby ulic	95
Odczyt danych wejściowych	98
Dlaczego korzystać z rekurencji?	105
Problem 2.: Odległość pomiędzy potomkami	105
Problem	105
Odczyt danych wejściowych	108
Liczba potomków w odległości d od wierzchołka	112
Liczba potomków dla wszystkich wierzchołków	114
Sortowanie wierzchołków	114
Wyświetlanie wyników	115
Funkcja main	116
Podsumowanie	117
Uwagi	118
3	
MEMOIZACJA I PROGRAMOWANIE DYNAMICZNE	119
Problem 1.: Burgerowa gorączka	120
Problem	120
Określenie planu rozwiązania problemu	120
Określanie optymalnego rozwiązania	122
Rozwiązanie 1.: Zastosowanie rekurencji	124

Rozwiązanie 2.: Memoizacja	129
Rozwiązanie 3.: Programowanie dynamiczne	135
Memoizacja i programowanie dynamiczne	138
Krok 1.: Struktura optymalnego rozwiązania	139
Krok 2.: Rozwiązanie rekurencyjne	140
Krok 3.: Memoizacja	140
Krok 4.: Programowanie dynamiczne	141
Problem 2.: Skąpcy	142
Problem	142
Określanie optymalnego rozwiązania	144
Rozwiązanie 1.: Rekurencja	146
Funkcja main	150
Rozwiązanie 2.: Memoizacja	152
Problem 3.: Rywalizacja hokejowa	154
Problem	154
Rozważania dotyczące rywalizacji	155
Określenie optymalnego rozwiązania	157
Rozwiązanie 1.: Rekurencja	160
Rozwiązanie 2.: Memoizacja	164
Rozwiązanie 3.: Programowanie dynamiczne	165
Optymalizacja zużycia pamięci	169
Podsumowanie	170
Uwagi	170
4	
ZAAWANSOWANA MEMOIZACJA I PROGRAMOWANIE DYNAMICZNE	171
Problem 1.: Skoczek	172
Problem	172
Analiza przykładu	173

Rozwiązanie 1.: Analiza wstecz	174
Rozwiązanie 2.: Analiza w przód	179
Problem 2.: Sposoby budowy	184
Problem	184
Analiza przykładu	185
Rozwiązanie 1.: Stosowanie „dokładnych” podproblemów	186
Rozwiązanie 2.: Dodawanie kolejnych podproblemów	191
Podsumowanie	196
Uwagi	196
5	
GRAFY I PRZESZUKIWANIE WSZERZ	197
Problem 1.: Pogoń skoczka	197
Problem	198
Optymalne ruchy skoczka	200
Najlepszy wynik skoczka	209
Przesunięcie i powrót skoczka	211
Optymalizacja czasu działania	215
Grafy i przeszukiwanie wszerek	216
Czym są grafy?	216
Grafy a drzewa	217
Algorytm BFS na grafach	218
Grafy a programowanie dynamiczne	220
Problem 2.: Wspinaczka po linie	221
Problem	221
Rozwiązanie 1.: Poszukiwanie ruchów	222
Rozwiązanie 2.: Nowy model	227
Problem 3.: Tłumaczenie książek	237
Problem	237

Wczytywanie nazw języków	238
Budowanie grafu	239
Implementacja algorytmu BFS	242
Koszt całkowity	244
Podsumowanie	245
Uwagi	245
6	
NAJKRÓTSZE ŚCIEŻKI NA GRAFACH WAŻONYCH	246
Problem 1.: Myszy w labiryncie	247
Problem	247
Zostawiamy algorytm BFS	248
Znajdowanie najkrótszej ścieżki na grafach ważonych	249
Tworzenie grafu	253
Implementacja algorytmu Dijkstry	255
Dwie optymalizacje	258
Algorytm Dijkstry	260
Efektywność działania algorytmu Dijkstry	260
Krawędzie o wagach ujemnych	261
Problem 2.: Planowanie odwiedzin u babci	264
Problem	264
Macierz sąsiedztwa	265
Konstruowanie grafu	266
Analiza przypadku testowego dziwacznych ścieżek	268
Zadanie 1.: Najkrótsze ścieżki	271
Zadanie 2.: Liczba najkrótszych ścieżek	273
Podsumowanie	281
Uwagi	281

WYSZUKIWANIE BINARNE	282
Problem 1.: Karmienie mrówek	283
Problem	283
Nowy rodzaj problemów z drzewami	285
Wczytywanie danych wejściowych	287
Sprawdzanie wykonalności	288
Poszukiwanie rozwiązania	291
Wyszukiwanie binarne	293
Wydajność działania algorytmu wyszukiwania binarnego	294
Określanie wykonalności	295
Przeszukiwanie tablicy posortowanej	295
Problem 2.: Skok przez rzekę	296
Problem	296
Koncepcja zachłanności	297
Testowanie wykonalności	299
Poszukiwanie rozwiązania	304
Wczytywanie danych wejściowych	307
Problem 3.: Jakość życia	308
Problem	308
Sortowanie wszystkich prostokątów	310
Wyszukiwanie binarne	313
Sprawdzanie wykonalności	314
Szybsze sprawdzanie wykonalności	316
Problem 4.: Drzwi w jaskini	322
Problem	323

Rozwiązywanie podzadań	324
Zastosowanie wyszukiwania liniowego	326
Stosowanie wyszukiwania binarnego	329
Podsumowanie	331
Uwagi	332
8	
KOPCE I DRZEWA SEGMENTÓW	333
Problem 1.: Promocja w supermarkecie	333
Problem	334
Rozwiązanie 1.: Wartość maksymalna i minimalna w tablicy	334
Kopce maksymalne	338
Kopce minimalne	351
Rozwiązanie 2.: Kopce	353
Kopce	356
Inne zastosowania	356
Wybór struktury danych	357
Problem 2.: Budowanie drzewców	358
Problem	358
Rekurencyjne wyświetlanie drzewców	360
Sortowanie na podstawie etykiet	361
Rozwiązanie 1.: Rekurencja	362
Pytania o sumę zakresu	365
Drzewa segmentów	367
Rozwiązanie 2.: Drzewa segmentów	376
Drzewa segmentów	377
Problem 3.: Suma dwóch	378
Problem	378
Wypełnianie drzewa segmentów	379

Znajdowanie odpowiedzi z użyciem drzewa segmentów	384
Aktualizacja drzewa segmentów	385
Funkcja main	389
Podsumowanie	389
Uwagi	390
9	
STRUKTURA ZBIORÓW ROZŁĄCZNYCH	391
Problem 1.: Sieć społecznościowa	392
Problem	392
Modelowanie danych w formie grafu	393
Rozwiązanie 1.: BFS	397
Struktura zbiorów rozłącznych	401
Rozwiązanie 2.: Struktura zbiorów rozłącznych	404
Optymalizacja 1.: Łączenie na podstawie wielkości	408
Optymalizacja 2.: Skracanie ścieżek	412
Struktura zbiorów rozłącznych	415
Relacje: Trzy wymagania	415
Wybieranie struktury zbiorów rozłącznych	415
Optymalizacje	416
Problem 2.: Przyjaciele i wrogowie	416
Problem	416
Rozszerzenie: Struktura zbiorów rozłącznych	418
Funkcja main	422
Operacje find i union	423
Operacje UstawJakoPrzyjaciół i UstawJakoWrogów	424
Operacje CzySąPrzyjaciółmi i CzySąWrogami	426
Problem 3.: Kłopot z szufladami	427
Problem	427
Równoważne szuflady	428
Funkcja main	434

Implementacja operacji find i union	436
Podsumowanie	437
Uwagi	437
10	
RANDOMIZACJA	438
Problem 1.: Yōkan	438
Problem	439
Losowy wybór kawałka	439
Generowanie liczb losowych	441
Określanie liczby kawałków	442
Odgadywanie smaków	445
Ilu prób potrzebujemy?	447
Wypełnianie tablic smaków	449
Funkcja main	450
Randomizacja	451
Algorytmy typu Monte Carlo	451
Algorytmy typu Las Vegas	453
Algorytmy deterministyczne a probabilistyczne	454
Problem 2.: Kapsle i butelki	454
Problem	454
Rozwiązywanie podzadania	455
Rozwiązanie 1.: Rekurencja	457
Rozwiązanie 2.: Dodanie randomizacji	461
Sortowanie szybkie	463
Implementacja algorytmu sortowania szybkiego	463
Efektywność najgorszego przypadku i oczekiwana	465
Podsumowanie	467
Uwagi	468
PODSUMOWANIE	469

A

EFEKTYWNOŚĆ ALGORYTMÓW	471
Kwestia czasu... i nie tylko	471
Notacja dużego O	473
Czas liniowy	473
Czas stały	475
Inny przykład	475
Czas kwadratowy	476
Notacja dużego O w tej książce	477

B

PONIEWAŻ NIE MOGŁEM SIĘ POWSTRZYMAĆ	479
Płatki śniegu: niejawne listy połączone	479
Burgerowa gorączka: rekonstrukcja rozwiązania	482
Pogoń skoczka: kodowanie ruchów	485
Algorytm Dijkstry: stosowanie kopca	487
Myszy w labiryncie: śledzenie z użyciem kopców	487
Myszy w labiryncie: implementacja z użyciem kopca	490
Skracanie skracania ścieżek	492
Krok 1.: Żadnych więcej operatorów trójargumentowych	492
Krok 2.: Bardziej czytelne operatory przypisania	493
Krok 3.: Wyjaśnienie rekurencji	494
Kapsle i butelki: sortowanie w miejscu	494

C

Z PODZIĘKOWANIEM ZA PROBLEMY	497
------------------------------------	-----