

Spis treści

PRZEDMOWA	13
PODZIĘKOWANIA	15
WPROWADZENIE	17
Zasoby internetowe	18
Dla kogo jest przeznaczona ta książka	18
Język programowania	19
Dlaczego wybrałem język C?	19
Słowo kluczowe static	19
Pliki nagłówkowe	20
Zwalnianie pamięci	21
Zagadnienia	21
Witryny oceniające	22
Anatomia opisu problemu	25
Problem: Kolejki po jedzeniu	26
Problem	26
Rozwiązanie problemu	27
Uwagi	29
1	
TABLICE MIESZAJĄCE	30
Problem 1. Płatki śniegu	30
Problem	30
Uproszczenie problemu	33
Rozwiązywanie podstawowego problemu	34
Rozwiązanie 1. Porównywanie parami	38
Rozwiązanie 2. Zmniejszenie liczby wykonywanych operacji	43
Tablice mieszające	49
Projekt tablicy mieszającej	49
Dlaczego warto używać tablic mieszających?	52

Problem 2. Słowa złożone	52
Problem	53
Wskazywanie słów złożonych	53
Rozwiązanie	54
Problem 3. Sprawdzanie pisowni — usuwanie litery	58
Problem	59
Rozważania o zastosowaniu tablic mieszających	60
Rozwiązanie doraźne	62
Podsumowanie	65
Uwagi	66
 2	
DRZEWA I REKURENCJA	67
Problem 1. Halloweenowy łup	67
Problem	68
Drzewa binarne	69
Rozwiązanie problemu dla przykładowego drzewa	71
Reprezentacja drzew binarnych	72
Zbieranie wszystkich cukierków	77
Zupełnie inne rozwiązanie	84
Przechodzenie minimalnej liczby ulic	90
Odczyt danych wejściowych	93
Dlaczego korzystać z rekurencji?	100
Problem 2. Odległość pomiędzy potomkami	101
Problem	101
Odczyt danych wejściowych	104
Liczba potomków w odległości d od wierzchołka	108
Liczba potomków dla wszystkich wierzchołków	110
Sortowanie wierzchołków	111
Wyświetlanie wyników	112
Funkcja main	113
Podsumowanie	114
Uwagi	114
 3	
MEMOIZACJA I PROGRAMOWANIE DYNAMICZNE	115
Problem 1. Burgerowa gorączka	115
Problem	116
Określenie planu rozwiązania problemu	116
Określanie optymalnego rozwiązania	118
Rozwiązanie 1. Zastosowanie rekurencji	120
Rozwiązanie 2. Memoizacja	126
Rozwiązanie 3. Programowanie dynamiczne	132

Memoizacja i programowanie dynamiczne	136
Krok 1. Struktura optymalnego rozwiązania	136
Krok 2. Rozwiązanie rekurencyjne	137
Krok 3. Memoizacja	137
Krok 4. Programowanie dynamiczne	138
Problem 2. Skąpcy	139
Problem	139
Określanie optymalnego rozwiązania	141
Rozwiązanie 1. Rekurencja	143
Funkcja main	148
Rozwiązanie 2. Memoizacja	149
Problem 3. Rywalizacja hokejowa	152
Problem	152
Rozważania dotyczące rywalizacji	153
Określenie optymalnego rozwiązania	155
Rozwiązanie 1. Rekurencja	158
Rozwiązanie 2. Memoizacja	161
Rozwiązanie 3. Programowanie dynamiczne	163
Optymalizacja zużycia pamięci	166
Problem 4. Sposoby zaliczenia	167
Problem	168
Rozwiązanie: memoizacja	168
Podsumowanie	170
Uwagi	170

4

GRAFY I PRZESZUKIWANIE WSZERZ

Problem 1. Pogoń skoczka	171
Problem	172
Optymalne ruchy skoczka	174
Najlepszy wynik skoczka	184
Przesunięcie i powrót skoczka	186
Optymalizacja czasu działania	189
Grafi i przeszukiwanie wszerek	190
Czym są grafi?	191
Grafi a drzewa	192
Algorytm BFS na grafach	194
Problem 2. Wspinaczka po linie	195
Problem	195
Rozwiązanie 1. Poszukiwanie ruchów	196
Rozwiązanie 2. Nowy model	202
Problem 3. Tłumaczenie książek	212
Problem	212
Budowanie grafu	213

Implementacja algorytmu BFS	217
Koszt całkowity	219
Podsumowanie	220
Uwagi	220
5	
NAJKRÓTSZE ŚCIEŻKI NA GRAFACH WAŻONYCH	221
Problem 1. Myszy w labiryncie	222
Problem	222
Zostawiamy algorytm BFS	223
Najkrótsze ścieżki na grafach ważonych	224
Tworzenie grafu	228
Implementacja algorytmu Dijkstry	231
Dwie optymalizacje	233
Algorytm Dijkstry	236
Efektywność działania algorytmu Dijkstry	236
Krawędzie o wagach ujemnych	237
Problem 2. Planowanie odwiedzin u babci	240
Problem	240
Macierz sąsiedztwa	241
Konstruowanie grafu	242
Dziwaczne ścieżki	244
Zadanie 1. Najkrótsze ścieżki	247
Zadanie 2. Liczba najkrótszych ścieżek	250
Podsumowanie	258
Uwagi	259
6	
WYSZUKIWANIE BINARNE	260
Problem 1. Karmienie mrówek	260
Problem	261
Nowy rodzaj problemów z drzewami	263
Wczytywanie danych wejściowych	264
Sprawdzanie wykonalności	266
Poszukiwanie rozwiązania	269
Wyszukiwanie binarne	271
Wydajność działania algorytmu wyszukiwania binarnego	272
Określanie wykonalności	273
Przeszukiwanie tablicy posortowanej	274
Problem 2. Skok przez rzekę	275
Problem	275
Koncepcja zachłanności	276
Testowanie wykonalności	278
Poszukiwanie rozwiązania	283
Wczytywanie danych wejściowych	287

Problem 3. Jakość życia	288
Problem	288
Sortowanie wszystkich prostokątów	290
Wyszukiwanie binarne	293
Sprawdzanie wykonalności	294
Szybsze sprawdzanie wykonalności	296
Problem 4. Drzwi w jaskini	303
Problem	303
Rozwiązanie podzadań	304
Zastosowanie wyszukiwania liniowego	306
Stosowanie wyszukiwania binarnego	309
Podsumowanie	312
Uwagi	312
7. PODZIĘKOWANIE ZA PROBLEMY	313
KOPCE I DRZEWIA SEGMENTÓW	313
Problem 1. Promocja w supermarkecie	313
Problem	314
Rozwiązanie 1. Wartość maksymalna i minimalna w tablicy	315
Kopce maksymalne	319
Kopce minimalne	332
Rozwiązanie 2. Kopce	334
Kopce	337
Inne zastosowania	337
Wybór struktury danych	338
Problem 2. Budowanie drzewców	339
Problem	339
Rekurencyjne wyświetlanie drzewców	342
Sortowanie na podstawie etykiet	343
Rozwiązanie 1. Rekurencja	344
Pytania o sumę zakresu	347
Drzewa segmentów	349
Rozwiązanie 2. Drzewa segmentów	359
Drzewa segmentów	360
Problem 3. Suma dwóch	361
Problem	361
Wypełnianie drzewa segmentów	362
Znajdowanie odpowiedzi z użyciem drzewa segmentów	367
Aktualizacja drzewa segmentów	369
Funkcja main	372
Podsumowanie	373
Uwagi	374

STRUKTURA ZBIORÓW ROZŁĄCZNYCH	375
Problem 1. Sieć społecznościowa	376
Problem	376
Modelowanie danych w formie grafu	377
Rozwiązanie 1. BFS	381
Struktura zbiorów rozłącznych	385
Rozwiązanie 2. Struktura zbiorów rozłącznych	389
Optymalizacja 1. łączenie na podstawie wielkości	393
Optymalizacja 2. Skracanie ścieżek	397
Struktura zbiorów rozłącznych	400
Relacje: Trzy wymagania	400
Wybieranie struktury zbiorów rozłącznych	401
Optymalizacje	401
Problem 2. Przyjaciele i wrogowie	401
Problem	402
Rozszerzenie: wrogowie	403
Funkcja main	408
Operacje find i union	409
Operacje UstawJakoPrzyjaciół i UstawJakoWrogów	410
Operacje CzySąPrzyjaciółmi i CzySąWrogami	412
Problem 3. Kłopot z szufladami	413
Problem	414
Równoważne szuflady	415
Funkcja main	421
Implementacja operacji find i union	423
Podsumowanie	424
Uwagi	424
PODSUMOWANIE	425
A	
EFEKTYWNOŚĆ ALGORYTMÓW	427
Kwestia czasu... i nie tylko	427
Notacja dużego O	429
Czas liniowy	429
Czas stały	431
Inny przykład	431
Czas kwadratowy	432
Notacja dużego O w tej książce	433

B		
PONIEWAŻ NIE MOGŁEM SIĘ POWSTRZYMAĆ		434
Płatki śniegu: niejawne listy połączone		434
Burgerowa gorączka: rekonstrukcja rozwiązań		438
Pogoń skoczka: kodowanie ruchów		440
Algorytm Dijkstry: stosowanie kopca		442
Myszy w labiryncie: śledzenie z użyciem kopców		443
Myszy w labiryncie: implementacja z użyciem kopca		446
Skracanie skracania ścieżek		447
Krok 1. Żadnych więcej operatorów trójargumentowych		448
Krok 2. Bardziej czytelne operatory przypisania		449
Krok 3. Wyjaśnienie rekurencji		449
C		
Z PODZIĘKOWANIEM ZA PROBLEMY		451

DLA POZACZKOWANEGO TENISTY JEDYNIEMIEJ TRUDNYM ZADANIEM JEST
 WYSTĄPIENIE PO GRZE W KOSCE (ZWYKLEJZA W WYPADEKU GRZY Z BEKHENDU). DO-
 NIEBNO WYNIKSTACZAK WYCIĄGAŁ, NI OGRANICZANU ZODSTAWIOWYCH UMIEJST-
 WOŚCI POWADZOWYCH WYJADAM, ZACZYNA SIĘ KŁOCIWAĆ PRZYJEMNOŚĆ Z GRZY I WALE-
 NIAŁOŚĆ SIŁKOWOŚĆ TENISA. POTEM KOSMA DZIAŁAŁ DO SWOJEGO REPERTARIU TENISU
 TO NOWEJZ MARYCZKA SIŁKA Z BEKHENDU, SIŁKOWE WYRAZNIKI, SKRÓTY... KOLEJNYM
 ETAPEM JEST NAPIĘCIE STRATEGICZNE NA WYŻSZYM POZIOMIE ABSTRAKCYJNE — GRZY
 SIŁKOWE I WALE, SIŁKA I SIŁKA, OBRONA Z GRZY SIŁKOWE. POCZYNAJĄC ROZWIJANY TITULOJ
 ZWIĄZKĄ Z WOLNOŚCIĄ NARZĘDZI I STRATEGII, KTORE BĄDĄ NAJLEPIEJ EFEKTYWNE W GRZE
 PRZECIWKO KONKRETNYM TYPIOM PRZECIWNIEKÓW — NIE OBIĘDZA JEDNO UNIWERSALNE
 ROZWIĄZANIE, KTORE SPRAWDZIŁOBY SIĘ W GRZE PRZECIWKO KAŻDEMU PRZECIWNIEKOWI.

Programowanie jest podobne do gry w tenisa. Dla początkujących programi-
 stów już samo skłonięcie komputera do zrealizowania, co chcą zrobić, jest dosta-
 tecznie trudnym zadaniem. Kiedy już opuszczają to pierwsze początkowe poziomy,
 zaczynają się prześledzić zabawy i frajda z rozwiązywania problemów: „w jaki spo-
 sób wynajść rozwiązanie danego problemu”. Choć nie istnieje żadna uniwer-
 salna metoda pozwalająca na efektywne rozwiązywanie wszystkich problemów
 obliczeniowych, to są dostępne niezależnie od technicznie zaawansowanego narzędzia
 i strategii: tablice mnożenia, drzewa wyszukiwania, rekurencja, memoizacja,
 programowanie dynamiczne, przewidywanie grałów itd. A dla wprawnych ob-
 wiadów problemów i algorytmów stanowi czerwiące włączanie i odnoszenie do tego,
 jakie narzędzia należy zastosować. Czy algorytm wyłownie powtarzające się wy-
 szkaldzanie, czy obliczanie wartości minimalnej? Jeśli tak, to możemy go przypie-
 czyć przy użyciu odpowiednio tablicy mieszającej lub kopca minimalnego! Czy