

Spis treści

PRZEDMOWA	13
PODZIĘKOWANIA	15
WPROWADZENIE	17

CZĘŚĆ I. ZROZUMIEĆ REKURENCJĘ 23

1	
CZYM JEST REKURENCJA?	25
Definicja rekurencji	26
Czym są funkcje?	28
Czym są stosy?	30
Czym jest stos wywołań?	32
Czym są funkcje rekurencyjne i przepełnienie stosu?	35
Przypadki bazowe i rekurencyjne	37
Kod przed wywołaniem rekurencyjnym i po wywołaniu rekurencyjnym	39
Podsumowanie	42
Materiały dodatkowe	43
Pytania praktyczne	43
2	
REKURENCJA A ITERACJA	44
Obliczanie silni	44
Iteracyjny algorytm obliczania silni	45
Rekurencyjny algorytm obliczania silni	46
Dlaczego rekurencyjny algorytm obliczania silni jest szalenie nieefektywny?	47
Znajdowanie wyrazów ciągu Fibonacciego	48
Iteracyjny algorytm wyznaczania n-tego wyrazu ciągu Fibonacciego	49
Rekurencyjny algorytm wyznaczania n-tego wyrazu ciągu Fibonacciego	50
Dlaczego rekurencyjny algorytm wyznaczania n-tego wyrazu ciągu Fibonacciego jest mocno nieefektywny?	52

Zamiana algorytmu rekurencyjnego na iteracyjny	53
Zamiana algorytmu iteracyjnego na rekurencyjny	55
Studium przypadku: obliczanie potęg	58
Rekurencyjna funkcja potęgująca	59
Iteracyjne obliczanie potęgi na podstawie wniosków z algorytmu rekurencyjnego	61
Kiedy powinno się korzystać z rekurencji?	64
Tworzenie algorytmów rekurencyjnych	66
Podsumowanie	67
Materiały dodatkowe	67
Pytania praktyczne	68
Zadania	68
3	
KLASYCZNE ALGORYTMY REKURENCYJNE	70
Dodawanie liczb zapisanych w tablicy	71
Odwracanie łańcucha znaków	74
Wykrywanie palindromów	78
Wieże Hanoi	80
Algorytm flood fill	86
Funkcja Ackermanna	91
Podsumowanie	94
Materiały dodatkowe	94
Pytania praktyczne	95
Zadania	96
4	
ALGORYTMY Z NAWROTAMI I ALGORYTMY PRZECHODZENIA PRZESZ DRZEWA	97
Algorytmy przechodzenia przeszech drzewo	98
Drzewa w Pythonie i JavaScriptcie	99
Przechodzenie przeszech drzewo	100
Przechodzenie przeszech drzewo w porządku preorder	101
Przechodzenie przeszech drzewo w porządku postorder	103
Przechodzenie przeszech drzewo w porządku inorder	104
Znajdowanie ośmioliterowych słów w drzewie	105
Ustalanie maksymalnej głębokości drzewa	108
Szukanie wyjścia z labiryntu	110
Podsumowanie	118
Materiały dodatkowe	119
Pytania praktyczne	119
Zadania	120

5	
ALGORYTMY TYPU „DZIEL I ZWYCIĘŻAJ”	121
Wyszukiwanie binarne — znajdowanie książki na półce z ułożonymi alfabetycznie pozycjami	122
Sortowanie szybkie — dzielenie nieposortowanej sterty książek na posortowane stosy	125
Sortowanie przez scalanie — łączenie małych stosów kart do gry w większe posortowane stosy	133
Sumowanie liczb zapisanych w tablicy	140
Algorytm mnożenia Karacuby	142
Matematyka kryjąca się za algorytmem Karacuby	149
Podsumowanie	150
Materiały dodatkowe	151
Pytania praktyczne	152
Zadania	153
6	
PERMUTACJE I KOMBINACJE	154
Podstawy teorii mnogości	155
Znajdowanie permutacji bez powtórzeń — usadzanie gości przy weselnym stole	157
Znajdowanie permutacji za pomocą zagnieżdżonych pętli — podejście dalekie od ideału	161
Permutacje z powtórzeniami — narzędzie do łamania haseł	164
Znajdowanie k-elementowych kombinacji za pomocą rekurencji	167
Znajdowanie wszystkich kombinacji zawierających poprawne nawiasowanie	173
Zbiór potęgowy — znajdowanie wszystkich podzbiorów zbioru	177
Podsumowanie	181
Materiały dodatkowe	182
Pytania praktyczne	183
Zadania	183
7	
MEMOIZACJA I PROGRAMOWANIE DYNAMICZNE	184
Memoizacja	184
Programowanie dynamiczne z zastosowaniem strategii top-down	185
Memoizacja w programowaniu funkcyjnym	186
Memoizacja w rekurencyjnym algorytmie wyznaczania elementów ciągu Fibonacciego	188
Moduł functools Pythona	192
Co się stanie, gdy przeprowadzimy memoizację „nieczystej” funkcji?	193
Podsumowanie	194
Materiały dodatkowe	195
Pytania praktyczne	195

OPTIMALIZACJA REKURENCJI OGONOWEJ	196
Jak działa rekurencja ogonowa i na czym polega jej optymalizacja?	197
Akumulatory w rekurencji ogonowej	198
Ograniczenia rekurencji ogonowej	200
Rekurencja ogonowa — studium przypadku	201
Rekurencja ogonowa — odwracanie łańcuchów znaków	201
Rekurencja ogonowa — znajdowanie podłańcuchów	203
Rekurencja ogonowa — potęgowanie	203
Rekurencja ogonowa — parzysty/nieparzysty	204
Podsumowanie	206
Materiały dodatkowe	206
Pytania praktyczne	207

RYŚOWANIE FRAKTALI	208
Grafika żółwia	209
Podstawowe funkcje modułu turtle	210
Trójkąt Sierpińskiego	213
Dywan Sierpińskiego	216
Drzewa fraktalne	220
Jak długie jest wybrzeże Wielkiej Brytanii? Krzywa i płatek śniegu Kocha	223
Krzywa Hilberta	227
Podsumowanie	230
Materiały dodatkowe	230
Pytania praktyczne	230
Zadania	231

CZĘŚĆ II. PROJEKTY 233

WYSZUKIWARKA PLIKÓW	235
Program do wyszukiwania plików	236
Funkcje dopasowujące	237
Znajdowanie plików, których rozmiar w bajtach jest parzysty	238
Znajdowanie plików, których nazwy zawierają każdą z pięciu samogłosek	239
Rekurencyjna funkcja walk()	239
Wywoływanie funkcji walk()	241
Funkcje biblioteki standardowej Pythona przydatne w pracy z plikami	242
Ustalanie nazwy pliku	242
Wyszukiwanie informacji o znacznikach czasowych pliku	243
Modyfikowanie plików	245
Podsumowanie	247
Materiały dodatkowe	247

11		
GENERATOR LABIRYNTÓW		248
Kod generatora labiryntów		249
Stałe w generatorze labiryntu		254
Tworzenie struktury danych labiryntu		255
Wyświetlanie struktury danych labiryntu		256
Korzystanie z rekurencyjnego algorytmu z nawrotami		258
Rozpoczynanie łańcucha wywołań rekurencyjnych		262
Podsumowanie		263
Materiały dodatkowe		263
12		
UKŁADANIE „PIĘTNASTKI”		264
Rekurencyjny algorytm układania „piętnastki”		265
Kod programu do układania „piętnastki”		267
Stałe w programie		276
Reprezentacja układanki w danych		277
Wyświetlanie układanki		277
Tworzenie nowej układanki		278
Znajdowanie współrzędnych pustego pola		279
Wykonywanie ruchu		280
Cofanie ruchu		281
Tworzenie nowej układanki		282
Rekurencyjne rozwiązywanie piętnastki		285
Funkcja solve()		285
Funkcja attemptMove()		287
Uruchamianie solvera		290
Podsumowanie		292
Materiały dodatkowe		292
13		
PROGRAM DO RYSOWANIA FRAKTALI		293
Fraktale dostępne w programie		293
Algorytm zastosowany w programie		295
Kod programu Fractal Art Maker		297
Stałe w programie i konfiguracja modułu turtle		301
Praca z funkcjami rysującymi kształty		301
Funkcja drawFilledSquare()		302
Funkcja drawTriangleOutline()		304
Funkcja drawFractal()		306
Początek funkcji		306
Obsługa słownika specyfikacji		307
Wykorzystywanie specyfikacji		309

Tworzenie przykładowych fraktali	312
Cztery rogi	312
Spirala kwadratów	312
Podwójna spirala kwadratów	313
Spirala trójkątów	313
Glider z „gry w życie” Conwaya	313
Trójkąt Sierpińskiego	314
Fala	314
Róg	315
Płatek śniegu	315
Rysowanie pojedynczego kwadratu lub trójkąta	316
Tworzenie własnych fraktali	316
Podsumowanie	317
Materiały dodatkowe	318

14

EFEKT DROSTE	319
Instalowanie biblioteki Pillow	320
Przygotowanie obrazka	320
Kod programu Droste Maker	322
Początek implementacji	324
Znajdowanie obszaru w kolorze magenty	325
Zmiana rozmiaru obrazka	327
Rekurencyjne umieszczanie obrazu w obrazie	330
Podsumowanie	332
Materiały dodatkowe	332