JACEK TOMASIEWICZ

ZAPRZYJAŹNIJ SIĘ Z ALGORYTMAMI

PRZEWODNIK DLA POCZĄTKUJĄCYCH I ŚREDNIO ZAAWANSOWANYCH

PWN

ZAPRZYJAŹNIJ SIĘ Z ALGORYTMAMI

Mojej mamie, Irenie Tomasiewicz,
oraz przyszłej żonie, Sylwii Stacheckiej

JACEK TOMASIEWICZ

ZAPRZYJAŹNIJ SIĘ Z ALGORYTMAMI

PRZEWODNIK DLA POCZĄTKUJĄCYCH I ŚREDNIO ZAAWANSOWANYCH

PWN

Projekt okładki i stron tytułowych **Hubert Zacharski**

Ilustracja na okładce **Ph_Dabphimsri/Shutterstock**

Wydawca **Łukasz Łopuszański**

Redaktor prowadzący **Jolanta Kowalczuk**

Koordynator produkcji **Anna Bączkowska**

Skład i łamanie **Fixpoint**

Książka, którą nabyłeś, jest dziełem twórcy i wydawcy. Prosimy, abyś przestrzegał praw, jakie im przysługują. Jej zawartość możesz udostępnić nieodpłatnie osobom bliskim lub osobiście znanym. Ale nie publikuj jej w internecie. Jeśli cytujesz jej fragmenty, nie zmieniaj ich treści i koniecznie zaznacz, czyje to dzieło. A kopiując jej część, rób to jedynie na użytek osobisty.

Szanujmy cudzą własność i prawo
Więcej na www.legalnakultura.pl
Polska Izba Książki

Copyright © by Wydawnictwo Naukowe PWN SA

Warszawa 2016

ISBN 978-83-01-18466-7

Wydanie I
Warszawa 2016

Wydawnictwo Naukowe PWN SA
02-460 Warszawa, ul. Gottlieba Daimlera 2
tel. 22 69 54 321, faks 22 69 54 288
infolinia 801 33 33 88
e-mail: pwn@pwn.com.pl, www.pwn.pl

Druk i oprawa: OSDW Azymut Sp. z o.o.

Spis treści

Wprowadzenie XI

Od autora XIII

1. Złożoność czasowa 1

Porównanie różnych złożoności czasowych 2
Limit czasu 3
Złożoność pamięciowa 4
Ćwiczenie 4
Zadania treningowe 5
Żabka ★ 5
Chodnik ★★ 6
Taśma ★★ 7
Rozwiązania 8

2. Zliczanie elementów 9

Ćwiczenie 10
Zadania treningowe 11
Permutacja ★ 11
Ropucha ★★ 12
Przyciski ★★★ 13
Rozwiązania 14

3. Sumy prefiksowe 16

Ćwiczenie 17
Zadania treningowe 18
Długa taśma ★ 18
Samochody ★★ 19

Chomiki ★★★ . . . 20
Rozwiązania . . . 21

4. Sortowanie . . . 22

Sposób 1: sortowanie przez wybieranie . . . 22
Sposób 2: sortowanie przez zliczanie . . . 23
Sposób 3: sortowanie przez scalanie . . . 24
Funkcje sortujące . . . 24
Ćwiczenie . . . 25
Zadania treningowe . . . 25
Iloczyn ★ . . . 25
Gwoździe ★★ . . . 26
Tory kolejowe ★★★ . . . 27
Rozwiązania . . . 28

5. Stos i kolejka . . . 30

Stos . . . 30
Kolejka . . . 31
Ćwiczenie . . . 32
Zadania treningowe . . . 32
Nawiasy ★★ . . . 32
Ryby ★★★ . . . 33
Cukiernia ★★★ . . . 34
Rozwiązania . . . 35

6. Wyszukiwanie lidera . . . 37

Sprawdzenie kandydata . . . 37
Rozwiązanie o złożoności $O(n^2)$ . . . 38
Rozwiązanie o złożoności $O(n \log n)$ . . . 38
Rozwiązanie o złożoności $O(n)$ . . . 39
Ćwiczenie . . . 40
Zadania treningowe . . . 41
Dwie części ★★ . . . 41
Bajtocka flaga ★★★ . . . 42
Lider prefiksowy ★★★ . . . 43
Rozwiązania . . . 44

7. Spójny podciąg o maksymalnej sumie . . . 45

Rozwiązanie o złożoności $O(n^3)$ . . . 45
Rozwiązanie o złożoności $O(n^2)$ . . . 46
Rozwiązanie o złożoności $O(n)$ . . . 46
Ćwiczenie . . . 47
Zadania treningowe . . . 49
Odchudzanie ★ . . . 49
Bilet ★★ . . . 49

Praca domowa ★★★ . . . 51
Rozwiązania . . . 51

8. Liczby pierwsze i złożone . . . 53

Liczenie dzielników . . . 53
Test pierwszości w czasie $O(\sqrt{n})$ . . . 54
Ćwiczenie . . . 54
Zadania treningowe . . . 56
Obwód ★ . . . 56
Szczyty ★★ . . . 56
Flagi ★★★★ . . . 57
Rozwiązania . . . 58

9. Sito Eratostenesa . . . 61

Faktoryzacja . . . 62
Ćwiczenie . . . 63
Zadania treningowe . . . 64
Tablica liczb ★★ . . . 64
Liczby półpierwsze ★★★ . . . 64
Liczby doskonałe ★★★ . . . 65
Rozwiązania . . . 66

10. Algorytm Euklidesa . . . 68

Najmniejsza wspólna wielokrotność . . . 69
Ćwiczenie . . . 69
Zadania treningowe . . . 69
Mandarynki ★ . . . 69
Wesoła małpka ★★ . . . 70
Zbiór pierwszych ★★★ . . . 71
Rozwiązania . . . 72

11. Ciąg Fibonacciego . . . 73

Ćwiczenie . . . 74
Zadania treningowe . . . 74
Zajączek ★★ . . . 74
Drabina ★★★ . . . 75
Spotkanie ★★★★ . . . 76
Rozwiązania . . . 77

12. Wyszukiwanie binarne . . . 79

Intuicja . . . 79
Implementacja . . . 80
Wyszukiwanie binarne po wyniku . . . 81
Ćwiczenie . . . 81
Zadania treningowe . . . 82

Promień ★★ . . . 82
Deski ★★★ . . . 82
Tort ★★★★ . . . 83
Rozwiązania . . . 85

13. Gąsienica . . . 87

Przykład użycia . . . 87
Ćwiczenie . . . 88
Zadania treningowe . . . 89
Smakołyki ★★ . . . 89
Wycinek ★★★ . . . 90
Temperatura ★★★★ . . . 90
Rozwiązania . . . 92

14. Programowanie zachłanne . . . 94

Problem wydawania reszty . . . 94
Dowodzenie poprawności . . . 95
Ćwiczenie . . . 95
Zadania treningowe . . . 96
Sznurki ★★ . . . 96
Bracia ★★★ . . . 97
Szklanki ★★★ . . . 98
Rozwiązania . . . 99

15. Programowanie dynamiczne . . . 101

Problem wydawania reszty . . . 101
Ćwiczenie . . . 103
Zadania treningowe . . . 104
Pionek ★ . . . 104
Wybrzeże ★★★ . . . 105
Ładny ciąg ★★★ . . . 106
Rozwiązania . . . 107

16. Drzewa binarne . . . 110

Pełne drzewo binarne . . . 111
Reprezentacja drzew binarnych . . . 111
Binarne drzewa wyszukiwania (BST) . . . 112
Ćwiczenie . . . 112
Zadania treningowe . . . 113
Drzewko binarne ★ . . . 113
Nieskończone drzewko ★★ . . . 114
Drzewko ★★★ . . . 115
Rozwiązania . . . 116

17. Kolejka priorytetowa . . . 119
Kopiec binarny . . . 119
Wstawienie elementu do kopca $O(\log n)$ . . . 120
Usunięcie elementu maksymalnego $O(\log n)$ . . . 121
Tworzenie nowego kopca z listy elementów $O(n)$ . . . 122
Ćwiczenie . . . 123
Zadania treningowe . . . 124
Emeryci ★★ . . . 124
Bilety ★★★ . . . 124
Tamy ★★★★ . . . 125
Rozwiązania . . . 127

18. Algorytmy grafowe BFS i DFS . . . 128
Rodzaje grafów . . . 129
Reprezentacja grafu . . . 130
DFS, czyli przeszukiwanie grafu w głąb . . . 132
BFS, czyli przeszukiwanie grafu wszerz . . . 133
Ćwiczenie . . . 134
Zadania treningowe . . . 134
Lista kontaktów ★★ . . . 134
Las ★★★ . . . 135
Wyprawa króla ★★★★ . . . 136
Rozwiązania . . . 137

A. Kolejne tematy . . . 139
Algorytm Dijkstry . . . 139
Średnica drzewa . . . 139
Zbiory rozłączne . . . 139
Algorytm Prima i Kruskala . . . 139
Sortowanie topologiczne . . . 140
Drzewo licznikowe . . . 140
Szybkie potęgowanie . . . 140
Koszt zamortyzowany . . . 140
Najdłuższy rosnący podciąg . . . 140
Teoria gier . . . 140
Algorytm Knutha–Morrisa–Pratta . . . 141
Haszowanie tekstów . . . 141
Algorytm Karpa–Millera–Rosenberga . . . 141
Szukanie palindromów i algorytm Manachera . . . 141
Najdłuższy wspólny podciąg . . . 141
Programowanie dynamiczne na drzewach . . . 141
Podstawy geometrii obliczeniowej . . . 142
Sortowanie kątowe . . . 142
Otoczka wypukła . . . 142
Para najmniej i najbardziej oddalonych punktów . . . 142

Maski bitowe . . . 142
Najniższy wspólny przodek . . . 142
Silnie spójne składowe . . . 143
Mosty i punkty artykulacji . . . 143
Cykl Eulera . . . 143
Przepływy i skojarzenia . . . 143

B. STL . . . 144

Para elementów . . . 144
Wektor . . . 145
Kolejka . . . 148
Kolejka priorytetowa . . . 149
Minimum, maksimum i zamiana . . . 150
Sortowanie . . . 150
Permutacje . . . 151
Mieszanie . . . 152
Wskaźniki w C++ . . . 152
Iteratory . . . 153
Wyszukiwanie binarne . . . 154
Lista . . . 154
Zbiór i multizbiór . . . 155
Mapa . . . 156

Bibliografia . . . 157

Skorowidz . . . 158

Opinie i komentarze . . . 161

Wprowadzenie

Komputer wkroczył niepostrzeżenie w życie każdego z nas. Prawie codziennie poszukujemy informacji w internecie, komunikujemy się ze znajomymi, wysyłając smsy, rozmawiamy przez Skype'a, kupujemy przez komputer bilety na pociąg lub do kina, bawimy się, grając na nim w gry. Z jednej strony stało się to możliwe dzięki rozwojowi technologii, która doprowadziła do miniaturyzacji urządzeń elektronicznych przy jednoczesnym zwiększaniu ich mocy obliczeniowej — procesory są coraz szybsze i są w stanie przechowywać oraz przetwarzać coraz więcej danych. Z drugiej strony komputer jest tylko urządzeniem elektronicznym o potencjalnie nieskończonych możliwościach, które nie znalazłyby tak powszechnego i szerokiego zastosowania, gdyby nie pomysłowość i umiejętności twórców programów komputerowych. Komputer jest w stanie wykonywać jedynie to, co nakaże mu programista za pomocą programu. Program komputerowy jest precyzyjnym zapisem algorytmu w języku, który jest zrozumiały dla komputera — w języku programowania. Algorytm zaś, potocznie, to precyzyjny przepis opisujący działanie, które może być wykonane przez człowieka, komputer lub inne medium, nie koniecznie rozumiejące istotę podjętego działania. Dziedzinę informatyki, która zajmuje się badaniem algorytmów, nazywa się *algorytmiką*.

Wybitny izraelski informatyk Dawid Harel w swojej znakomitej książce *Rzecz o istocie informatyki. Algorytmika* napisał: *Przepisy określiliśmy mianem algorytmów, obszar zaś ludzkich dociekań, wiedzy i doświadczeń dotyczących algorytmów nazwiemy algorytmiką.* I dalej: *Algorytmika to więcej niż dział informatyki. Tkwi ona w centrum wszystkich działów informatyki.* W tym miejscu przypomnijmy jeszcze słowa Donalda Knutha (pierwszego badacza, który zajął się systematycznym badaniem algorytmów) z jego trylogii *Sztuka programowania*: *Proces przygotowywania programu dla komputera cyfrowego jest pociągający nie tylko ze względu na potencjalne korzyści ekono-*

miczne czy naukowe. Wiążą się z nim również przeżycia estetyczne, podobne do tworzenia poezji lub komponowania muzyki.

Książka Jacka Tomasiewicza jest znakomitym wprowadzeniem w świat algorytmów i programowania dla początkujących i średnio zaawansowanych, którzy chcą osiągnąć mistrzostwo w najbardziej podstawowym dziale informatyki, jakim jest algorytmika. Znakomicie dobrane zadania i odpowiednie stopniowanie trudności pozwalają początkującym informatykom zapoznać się z podstawowymi metodami projektowania i analizowania algorytmów ze szczególnym uwzględnieniem ich złożoności obliczeniowej. Autor zebrał zadania, które z powodzeniem wykorzystywał na zajęciach z młodymi adeptami algorytmiki ze szkół ponadgimnazjalnych i gimnazjów. Jako młody człowiek, finalista Olimpiady Informatycznej, doskonale rozumie, co jest najtrudniejsze w poznawaniu algorytmiki na początkowym etapie. Dlatego stara się trudne tematy tłumaczyć bez specjalnego formalizmu, językiem w pełni zrozumiałym nawet dla gimnazjalisty.

Książka Jacka Tomasiewicza jest godna polecenia nie tylko wszystkim zainteresowanym algorytmiką i programowaniem. Będzie także pomocna w powszechnym nauczaniu algorytmiki w gimnazjach i szkołach ponadgimnazjalnych. Zbioru zadań z przystępnymi zadaniami algorytmicznymi bardzo brakowało. Gorąco polecam książkę autorstwa Jacka Tomasiewicza!

prof. dr hab. Krzysztof Diks

Przewodniczący Komitetu Głównego Olimpiady Informatycznej
Instytut Informatyki Uniwersytetu Warszawskiego

Od autora

Programowanie w dzisiejszych czasach staje się coraz bardziej popularne. Jest na tyle ważną umiejętnością, że jego nauczanie zaczyna się od szkół gimnazjalnych lub nawet podstawowych. W programach nauczania jest coraz więcej algorytmiki. Pojawiają się takie tematy, jak rozwiązywanie problemów z zastosowaniem podejścia algorytmicznego. Udział dużej liczby uczestników w Olimpiadzie Informatycznej dedykowanej uczniom szkół ponadgimnazjalnych oraz w Olimpiadzie Informatycznej Gimnazjalistów świadczy o popularności programowania wśród młodych osób.

Początki programowania i pierwsze próby rozwiązywania zadań algorytmicznych mogą być jednak bardzo trudne. Wiele osób poddaje się już po kilku nieudanych podejściach rozwiązania złożonych problemów. Istnieje wiele książek poświęconych analizie zaawansowanych algorytmów, a brakuje materiałów dla początkujących i średnio zaawansowanych oraz książek dla nauczycieli, którzy są bardzo ważni w procesie kształcenia przyszłych programistów.

Książka ta jest swego rodzaju wstępem do algorytmiki oraz podręcznikiem nauczania. Nie zawiera trudnych, zaawansowanych algorytmów, których opis można znaleźć w wielu innych źródłach. Zawiera za to opis i implementację najważniejszych, prostych algorytmów oraz zbiór zadań umożliwiających przećwiczenie ich w praktyce.

Grupą docelową książki są nie tylko uczniowie rozpoczynający swoją przygodę z algorytmiką, ale także nauczyciele informatyki poszerzający swoją wiedzę algorytmiczną. Książka ma szansę wpłynąć pozytywnie na jakość kształcenia uczniów oraz doprowadzić do zwiększenia liczby osób startujących w konkursach informatycznych.

Pomysł na książkę narodził się podczas organizacji konkursów, kół naukowych oraz obozów informatycznych na różnym poziomie zaawansowania. Największym wyzwaniem było wymyślenie prostych, lecz nietrywialnych

zadań oraz przekazanie wiedzy osobom początkującym. W książce zostały umieszczone najciekawsze i najbardziej odpowiednie zadania zbierane i przygotowywane przez autora od momentu rozpoczęcia studiów. Najważniejsze tematy zostały wybrane na podstawie doświadczenia z ośmiu lat nauczania.

W przypadku jakichkolwiek pytań dotyczących książki prosimy o kontakt do autora na adres mailowy `programernia@gmail.com`. Autor prowadzi również zajęcia z programowania. Jeśli chciałbyś się dowiedzieć więcej na temat tych zajęć, skontaktuj się, pisząc na podany adres.

Struktura książki

Książka składa się z 18 rozdziałów, w których są omawiane algorytmy oraz techniki programowania stosowane w początkach nauczania algorytmiki. Rozdziały zostały ułożone chronologicznie i poruszają tematy z różnych dziedzin, takich jak struktury danych, teoria liczb i teoria grafów.

Każdy rozdział został podzielony na trzy części. W pierwszej z nich w krótki, przystępny sposób jest opisana wybrana technika programowania lub algorytm wraz z jego implementacją, aby zaznajomić Czytelnika z omawianym tematem. Druga część to ćwiczenie, które porusza problematykę zagadnienia i prezentuje sposób wykorzystania algorytmu w zadaniach. W każdym ćwiczeniu został przedstawiony dokładny opis rozwiązania problemu. Trzecia część zawiera zestaw trzech zadań treningowych o rosnącym stopniu trudności, dzięki którym Czytelnik może sprawdzić nabyte umiejętności w praktyce. Dodatkowo, do każdego zadania zostały przygotowane wskazówki — po kilka podpowiedzi, które mogą być pomocne podczas rozwiązywania zadania. Elementy implementacji algorytmów oraz rozwiązań zadań zapisane są w popularnym obecnie języku C++.

W dodatku A znajduje się lista 26 bardziej zaawansowanych algorytmów i struktur danych (z krótkim opisem), ułożonych od najprostszych do najtrudniejszych. W dodatku B jest podany opis standardowej biblioteki wzorców (STL), czyli zbioru bibliotek w C++ przydatnych w implementacji algorytmów.

Ćwiczenia i zadania treningowe

Poszczególne rozdziały książki zawierają zadania związane z omawianą tematyką (łącznie 18 ćwiczeń i ponad 50 zadań treningowych). Wiedza potrzebna do rozwiązania kolejnych zadań znajduje się w rozdziałach je poprzedzających. Dzięki temu Czytelnik, po przyswojeniu materiału zawartego w kolejnych

rozdziałach, będzie w stanie rozwiązać wszystkie zadania. Większość z nich pochodzi z ogólnopolskich konkursów informatycznych, takich jak Olimpiada Informatyczna lub Potyczki Algorytmiczne, oraz z obozów informatycznych ILOCAMP. Zadania zostały podzielone na cztery kategorie i są oznaczone minimalnie jedną (zadania proste), a maksymalnie czterema gwiazdkami (zadania bardzo trudne).

W zadaniach prostych oraz o średniej trudności do rozwiązania wystarczy podstawowa znajomość pewnego algorytmu oraz umiejętność jego praktycznego zastosowania. Charakteryzują się wysokim odsetkiem poprawnych rozwiązań przez uczestników konkursów, w których zadania zostały użyte, oraz łatwością zrozumienia rozwiązania. Zadania trudne wymagają dodatkowego pomysłu. Co więcej, mogą wymagać zastosowania algorytmów z poprzednich rozdziałów oraz cechują się bardziej złożoną implementacją. Zadania bardzo trudne zawierają złożone problemy, w których wykorzystanie algorytmów nie jest łatwo dostrzegalne. Charakteryzują się małym odsetkiem poprawnych i optymalnych rozwiązań przez uczestników. Przypisanie poziomu trudności bazuje na doświadczeniu autora w pracy z uczniami oraz na statystykach z przeprowadzonych konkursów.

Testowanie rozwiązań

Rozwiązania zadań treningowych można przetestować w serwisie na stronie internetowej `http://main2.edu.pl`. Po wybraniu odpowiedniego kursu `http://main2.edu.pl/c/zadania-pwn` i zadania wysłany kod zostanie przetestowany pod względem poprawności oraz wydajności.

Wymagania wstępne

Książka dotyczy algorytmiki, a nie konkretnego języka programowania. Wszystkie przedstawione w książce algorytmy są zaimplementowane w języku C++, którego składnię Czytelnik może poznawać, czytając kolejne rozdziały. Początkowo wystarczy znajomość podstaw jednego języka programowania, takich jak warunki, pętle i tablice.

Konwencje

Przedstawione implementacje zakładają, że:

- wszystkie zmienne są początkowo wyzerowane,
- pomijamy wczytywanie i wypisywanie danych oraz załączanie bibliotek,

- elementy w tablicach są umieszczone od indeksu 0, chyba że napisano inaczej.

Metody pracy

Książka może być wykorzystywana przez uczniów do samodzielnej nauki podstawowych algorytmów oraz przez osoby chcące poszerzyć swoją wiedzę algorytmiczną. Może też być doskonałym podręcznikiem dla nauczycieli uczących w szkołach lub dla wykładowców przeprowadzających dodatkowe zajęcia informatyczne.

Poniżej przedstawiono metody pracy dla osób dokształcających się samodzielnie oraz dla nauczycieli przekazujących swoją wiedzę innym.

Do uczniów

Naukę algorytmiki warto rozplanować w czasie. Przykładowo, można się zobowiązać do przerabiania jednego rozdziału tygodniowo. Wszystkie rozdziały mają taką samą formę, warto więc zapoznać się z opisem pracy z pojedynczego rozdziału.

1. Na początku przeczytaj ze zrozumieniem opis algorytmu lub techniki programowania (całość do części z ćwiczeniem). W opisach często znajdziesz implementacje, które warto dokładnie prześledzić.
2. Druga część to ćwiczenie. Zanim przeczytasz rozwiązanie, spróbuj je wymyślić samodzielnie. Wystarczy ogólny pomysł na rozwiązanie, bez implementacji. Następnie opisane rozwiązanie porównaj z wymyślonym przez siebie i ostatecznie je zaimplementuj (abyś był pewien, że zrozumiałeś algorytm).
3. Trzecia część to zadania treningowe ułożone od najprostszych do najtrudniejszych. Po wymyśleniu optymalnego rozwiązania spróbuj je zaimplementować i przetestować pod względem poprawności i wydajności. W przypadku problemów z pomysłem na algorytm, na końcu rozdziału znajdziesz wskazówki ułatwiające wymyślenie rozwiązania. Po przeczytaniu każdej z nich spróbuj ponownie podejść do zadania. Jeśli po przeczytaniu wszystkich nie będziesz w stanie zrozumieć ani zaimplementować rozwiązania, to odłóż zadanie na później i przejdź do kolejnego rozdziału (ale tylko w przypadku, gdy rozwiążesz co najmniej jedno zadanie). Każdy rozdział zawiera przynajmniej jedno trudne zadanie, dlatego nie przejmuj się w przypadku nieudanych prób jego rozwiązania.

Zawsze warto dokumentować postępy w nauce. Przykładowo, możesz utworzyć tabelę i zaznaczać, które rozdziały już przeczytałeś i jakie zadania

udało Ci się rozwiązać. Takie postępowanie motywuje do dalszej pracy oraz pozwala śledzić postępy w nauce.

Po przerobieniu wszystkich rozdziałów możesz przejść do kolejnych, bardziej zaawansowanych tematów. W dodatku B znajdziesz listę 26 najważniejszych algorytmów. Dokładniejszy opis tych zagadnień możesz znaleźć w wielu informatycznych książkach, takich jak *Algorytmy i struktury danych* [BDR] czy *Wprowadzenie do algorytmów* [WDA].

Do nauczycieli

Poniżej znajduje się opis pracy dla nauczyciela uczącego grupę osób lub przeprowadzającego indywidualne korepetycje. Początkowo warto zaznajomić się ze wszystkimi rozdziałami i rozplanować nauczanie w czasie. W miarę potrzeb można wprowadzić kilka zagadnień, które nie zostały opisane w książce. Mogą to być tematy poprzedzające (takie jak nauka programowania) lub tematy pomiędzy rozdziałami (takie jak rekurencja, systemy liczbowe lub szyfrowanie).

Pierwsza część to przedstawienie uczniom zagadnień teoretycznych, czyli omówienie algorytmu i pokazanie implementacji. Jest to bardzo ważny fragment. Należy wówczas kontrolować, czy uczniowie nadążają z przyswajaniem materiału.

Druga część to ćwiczenie teoretyczno-praktyczne, w którym pomysł na rozwiązanie będzie szukany wspólnie z nauczycielem. Nauczyciel pokazuje ćwiczenie i daje uczniom trochę czasu na samodzielne zastanowienie się nad problemem. Następnie chętne osoby mogą przedstawić swoje spostrzeżenia, a nauczyciel naprowadza grupę na rozwiązanie. W tej części ważne jest pokazanie procesu wymyślania rozwiązania. Na koniec, celem pełnego zrozumienia, uczniowie powinni zaimplementować rozwiązanie.

Trzecia część to zadania treningowe, które sprawdzają praktyczną wiedzę uczniów. Wybór zadań powinien być dostosowany do poziomu grupy. W przypadku słabszych grup i ograniczonego czasu należy zadać wyłącznie najprostsze zadanie, natomiast w przypadku mocniejszych grup warto dać wszystkie zadania. Rozwiązywanie i implementowanie może odbywać się w trakcie tak zwanego laboratorium, w którym nauczyciel może rozważnie pomagać uczniom. Zadania mogą być również zadane jako praca domowa. Zawsze po upływie terminu na rozwiązywanie zadań należy przedstawić całej grupie dokładną metodę rozwiązania (najlepiej w formie podobnej do ćwiczeń) oraz zachęcić uczniów do implementacji rozwiązań.

Po przerobieniu wszystkich rozdziałów nauczyciel może przejść do bardziej zaawansowanych algorytmów. Propozycja listy takich zagadnień znajduje się w dodatku *B*.