



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 doksztalających 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 rozumiałeś 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.

Podziękowania

Książka ta powstała na podstawie mojej pracy magisterskiej, którą obroniłem w czerwcu 2015 roku na Uniwersytecie Warszawskim. Praca była pisana w latach 2013–2014, jednak materiały do niej powstawały również w latach wcześniejszych. Ogromnie dziękuję opiekunowi pracy, dr. Jakubowi Radoszewskiemu, za cenne uwagi merytoryczne, poświęcony czas oraz wielkie zaangażowanie.

Dziękuję również Grzegorzowi Jakackiemu — szefowi firmy Codility — za wsparcie oraz umożliwienie jednoczesnego pogodzenia pracy zawodowej oraz pisania pracy magisterskiej. Wielkie podziękowania należą się także dr. Marcinowi Kubicy, dr. Tomaszowi Idziaszkowi oraz Przemysławowi Jakubowi Kozłowskiemu za cenne wskazówki oraz uwagi merytoryczne dotyczące pracy.

Jednym z wyzwań podczas gromadzenia materiałów do pracy był proces wybierania odpowiednich zadań. Chciałbym podziękować Joannie Bujnowskiej za udostępnienie swoich zadań *Wesoła matka* i *Drzewko*, dr. Jakubowi Radoszewskiemu za udostępnienie zadań *Liczby doskonałe* i *Spotkanie* oraz dr. Marcinowi Kubicy za udostępnienie zadania *Obwód*.

Podziękowania kieruję również do prof. Krzysztofa Diksa — świetnego wykładowcy oraz przewodniczącego Komitetu Głównego Olimpiady Informatycznej — za merytoryczne wsparcie oraz wszelaką pomoc. Dziękuję też Ireneuszowi Bujnowskiemu — znakomitemu nauczycielowi, który zaszczepił we mnie pasję do informatyki.