Bez **algorytmiki** nie ma informatyki!

Piotr Stańczyk

Algorytmika praktyczna

Nie tylko dla mistrzów

PWN

ALGORYTMIKA PRAKTYCZNA

NIE TYLKO DLA MISTRZÓW

PIOTR STAŃCZYK

ALGORYTMIKA PRAKTYCZNA

NIE TYLKO DLA MISTRZÓW

PWN

MIKOM

Projekt okładki: **Anna Kozłowska KA PROJEKT**

Redakcja: **Ewa Zdanowicz**

Skład komputerowy: **Piotr Stańczyk**

Wszelkie uwagi na temat książki prosimy kierować pod adresem poczty elektronicznej Autora: stanczyk@mimuw.edu.pl

ISBN 978-83-01-15821-7

Wydanie I – 2 dodruk
Warszawa 2019

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; reklama@pwn.pl
www.pwn.pl

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

Spis treści

Słowo wstępne

Bez algorytmiki nie ma informatyki. Każdy, kto chce zostać rasowym informatykiem, powinien poświęcić dużo czasu na projektowanie, analizowanie i programowanie algorytmów. Oddawana właśnie do rąk czytelników książka Piotra Stańczyka *Algorytmika praktyczna. Nie tylko dla mistrzów* może pomóc w doskonaleniu sztuki rozwiązywania problemów algorytmicznych i biegłego ich zapisywania w postaci eleganckich, wydajnie działających programów. Jest to książka dla osób, które poznały już podstawy programowania oraz algorytmiki i chcą rozwijać swoje umiejętności na tym polu poprzez naukę na dobrych, sprawdzonych przykładach. Znajdą one tutaj omówienie szeregu zagadnień współczesnej algorytmiki, obejmujących algorytmy grafowe, geometrię obliczeniową, kombinatorykę, teorię liczb, struktury danych, algorytmy tekstowe i algorytmy algebry liniowej. Autor podaje minimalną liczbę pojęć i faktów potrzebnych do zrozumienia prezentowanych technik algorytmicznych i algorytmów, a same algorytmy i techniki przedstawia na przykładzie rozwiązań wybranych zadań z konkursów programistycznych. Niezwykle cenne jest to, że każde rozwiązanie zadania, to bardzo czytelny program gotowy do uruchomienia.

W niniejszej książce Piotr Stańczyk dzieli się swoimi i kolegów doświadczeniami z udziału w konkursach algorytmicznych. Są to doświadczenia osób, które potwierdziły swoją wiedzę i umiejętności algorytmiczno-programistyczne w konfrontacji z rówieśnikami z całego świata. Piotr to dwukrotny laureat krajowej Olimpiady Informatycznej oraz dwukrotny medalista Bałtyckiej Olimpiady Informatycznej. Wraz z kolegami zwyciężał w Akademickich Mistrzostwach Polski w Programowaniu Zespołowym i zdobywał medale w Akademickich Mistrzostwach Europy Środkowej. Piotra niewątpliwie można zaliczyć do polskiej szkoły informatyki młodzieżowej, której przedstawiciele dwukrotnie odnosili zwycięstwo w Międzynarodowej Olimpiadzie Informatycznej (Filip Wolski w 2006 roku oraz Tomasz Kulczyński w 2007 roku) oraz zwyciężali w prestiżowych Akademickich Mistrzostwach Świata w Programowaniu Zespołowym (zespoły Uniwersytetu Warszawskiego w składach: Tomasz Czajka, Andrzej Gąsienica-Samek, Krzysztof Onak — zwycięstwo w 2003 roku; Marek Cygan, Marcin Pilipczuk, Filip Wolski — zwycięstwo w 2007 roku).

Książka Piotra Stańczyka powinna znaleźć się na półce każdego zainteresowanego odnoszeniem sukcesów w konkursach algorytmicznych. Ale nie tylko. Jestem przekonany, że może ona być wykorzystywana na zajęciach z programowania oraz algorytmów i struktur danych zarówno dla uzdolnionej młodzieży szkół średnich, jak i studentów studiów informatycznych. Czytelnikom życzę dużo przyjemności z korzystania z *Algorytmiki praktycznej*, jak też zachęcam do krytycznego czytania i dzielenia się uwagami z autorem. Piotrowi życzę satysfakcji z wykonanej pracy, ale także wytrwałości w kontaktach z czytelnikami i uwzględniania wszystkich konstruktywnych uwag w kolejnych wydaniach.

Krzysztof Diks

Przedmowa

Pierwsza polska Olimpiada Informatyczna dla uczniów szkół średnich została zorganizowana w 1993 roku przez największe polskie uczelnie, takie jak Uniwersytet Warszawski, Uniwersytet Jagielloński czy Uniwersytet Wrocławski. Zwycięzcy Olimpiad Informatycznych mają możliwość reprezentowania Polski podczas takich międzynarodowych konkursów jak Olimpiada Informatyczna Krajów Bałtyckich, Olimpiada Informatyczna Centralnej Europy i — najbardziej prestiżowa — Międzynarodowa Olimpiada Informatyczna (strona konkursu *http://www.ioinformatics.org/*). Coraz większa z roku na rok liczba uczestników tych olimpiad świadczy o rosnącej popularności nauki algorytmiki wśród uczniów szkół średnich. W 2006 roku rozpoczęto organizowanie Olimpiady Informatycznej Gimnazjalistów. Swego rodzaju „olimpiadą informatyczną" dla studentów są Akademickie Mistrzostwa Polski w Programowaniu Zespołowym. Biorą w nich udział reprezentacje polskich uczelni, wyłaniane w lokalnych konkursach eliminacyjnych.

Mistrzostwa Polski otwierają drogę ku najbardziej prestiżowemu i cieszącemu się długą tradycją konkursowi, jakim jest International Collegiate Programming Contest organizowany przez Association for Computing Machinery (w skrócie ACM; oficjalna strona konkursu *http://icpc.baylor.edu/icpc/*). Pierwsze zawody finałowe odbyły się 2 lutego 1977 roku w Atlancie, w Stanach Zjednoczonych. Na początku lat dziewięćdziesiątych w eliminacjach do tego konkursu brało udział około 400 drużyn, z których 25 kwalifikowało się co roku do finału. Obecnie o 70–80 miejsc w finałach walczy ponad 4000 drużyn z 1600 uczelni całego świata.

Oprócz tych cieszących się wieloletnią tradycją konkursów jest wymyślanych i organizowanych wiele nowych — zarówno polskich, takich jak Potyczki Algorytmiczne czy Internetowy Turniej Programów Walczących, jak i międzynarodowych, do których należą TopCoder, Google Code Jam czy Microsoft Imagine Cup. Branie udziału w konkursach jest zarówno dla uczniów, jak i studentów ważnym elementem nauki algorytmiki.

Poziom konkursów nieustannie się podwyższa. Opracowywane są kolejne metody przygotowywania się do zawodów i specjalne techniki pisania programów. Ich celem jest zminimalizowanie czasu potrzebnego na rozwiązanie zadania, przy jednoczesnym unikaniu możliwie dużej liczby błędów. Ze względu na ograniczony czas trwania konkursów ich uczestnicy nie mogą rozwiązywać wszystkich zadań od zera. W wielu przypadkach muszą wykorzystywać pomysły zastosowane w podobnych zadaniach rozwiązywanych już wcześniej, co pozwala im zaoszczędzić trochę czasu.

Książka ta ma służyć do nauki algorytmiki, nie jest jednak typowym podręcznikiem. Nie zawiera wnikliwego opisu algorytmów wraz z ich analizą złożoności oraz dowodem poprawności. Zamiast tego obejmuje kolekcję implementacji różnych algorytmów bardzo przydatnych podczas zawodów i zbiór zadań umożliwiający przećwiczenie ich wykorzystania w praktyce.

Znaczącą część grona czytelników stanowić będą zapewne osoby zainteresowane zwiększeniem swoich umiejętności w zakresie szybkiego implementowania rozwiązań zadań algorytmicznych w języku C++. Lektura książki na pewno pozwoli na szybsze rozwiązywanie zadań podczas takich konkursów jak Olimpiada Informatyczna. Użyteczność tej

publikacji jest szczególnie znacząca, gdy bierze się udział w konkursach typu ACM ICPC, w trakcie których jest dozwolone korzystanie z literatury. Wiele algorytmów z tej książki może być po prostu przepisanych do implementowanych podczas zawodów programów. Prezentowane algorytmy zostały tak napisane, aby ich adaptacja — w celu wykorzystania w zadaniach — była jak najprostsza, a jednocześnie ich kod źródłowy jak najkrótszy.

Pozycja ta może służyć nie tylko zawodnikom — jej lektura w połączeniu z lekturą książki dotyczącej podstaw algorytmiki, takiej jak *Wprowadzenie do algorytmów* [4], może stanowić także dla innych osób doskonałą metodę nauki tej dyscypliny „od podstaw", łączącą teoretyczną analizę algorytmów z podejściem bardziej praktycznym, opisanym w tej książce. Naukę taką w istotny sposób ułatwią liczne odwołania do literatury.

Pomysł napisania książki narodził się podczas treningów zespołu (autora) *Warsaw Predators* do zawodów ACM ICPC. W czasie wielu sesji treningowych była tworzona i rozbudowywana biblioteczka algorytmiczna, w której były umieszczane implementacje najczęściej wykorzystywanych w trakcie zawodów algorytmów i struktur danych. Książkę tę można określić mianem szczegółowej dokumentacji do biblioteczki algorytmicznej drużyny *Warsaw Predators*. Należy mieć nadzieję, że materiały w niej zawarte pomogą wielu osobom w przygotowaniach do zawodów. Powodzenia!

Struktura książki

Książka jest podzielona na osiem rozdziałów. W pierwszych siedmiu są omówione algorytmy z różnych dziedzin algorytmiki: teorii grafów, geometrii obliczeniowej, kombinatoryki, teorii liczb, struktur danych, algorytmów tekstowych oraz algebry liniowej. Każdy z tych rozdziałów zawiera różne algorytmy wraz z ich implementacjami, pochodzącymi z biblioteczki algorytmicznej. Ich omówienie ma na celu przybliżenie problematyki poruszanego zagadnienia i zaprezentowanie sposobu wykorzystania algorytmów w zadaniach. Aby zapoznać się z dowodami poprawności bądź dokładną analizą złożoności, należy sięgać do literatury, do której odwołania są umieszczone w różnych miejscach książki. Ostatni, ósmy, rozdział jest poświęcony elementom strategii podczas zawodów.

Po zasadniczej części pracy następują przydatne dodatki. W dodatku A są podane nagłówki stosowane w programach, a dodatek B zawiera nagłówki Eryka Kopczyńskiego na konkurs TopCoder. Z kolei w dodatku C są zebrane obserwacje oraz rady pochodzące od wielu światowej klasy zawodników. Lektura tego dodatku może okazać się niezwykle cenna, gdyż zawarte w nim informacje pomagają uniknąć wielu problemów podczas przygotowań do zawodów. Następne dwa dodatki dotyczą programowania dynamicznego (dodatek D) oraz zachłannego (dodatek E). Ponieważ z tymi dwiema technikami nie wiążą się specyficzne struktury danych ani algorytmy (w rozwiązaniach zadań opartych na programowaniu dynamicznym lub zachłannym wykorzystuje się algorytmy z różnych dziedzin), zatem dodatki te nie zawierają żadnych przydatnych implementacji, a jedynie wykazy wielu zadań, na których można ćwiczyć umiejętność stosowania tych metod w praktyce.

W poszczególnych rozdziałach książki zamieszczone są przykładowe zadania związane z omawianą tematyką (wykaz tych zadań znajduje się w dodatku F). Algorytmy potrzebne do ich rozwiązania są opisane, w miarę możliwości, w rozdziałach poprzedzających wystąpienie kolejnych zadań. Dzięki temu, śledząc po kolei materiał zawarty w książce, czytelnik będzie posiadał wiedzę potrzebną do rozwiązania wszystkich zadań. Większość z nich pochodzi z polskich konkursów informatycznych, takich jak Olimpiada Informatyczna, Potyczki Algorytmiczne czy Pogromcy Algorytmów. Programy stanowiące ich rozwiązania można pobrać ze strony internetowej niniejszej książki w księgarni

PWN (*http://ksiegarnia.pwn.pl/*) — przy każdym zadaniu jest podana nazwa pliku zawierającego rozwiązanie. Ponadto w końcowej części książki są umieszczone wskazówki do wszystkich tych zadań. Do każdego z nich jest po kilka podpowiedzi, które mogą być pomocne podczas jego rozwiązywania — każda kolejna wskazówka uściśla pomysł naszkicowany w poprzedniej. Takie podejście zapewnia, że jeśli początkowe wskazówki okażą się niewystarczające, można skorzystać z następnych.

Oprócz przykładowych pełnych zadań w książce można znaleźć także wykazy zadań pochodzących z internetowych serwisów umożliwiających automatyczną weryfikację poprawności nadsyłanych rozwiązań. Czytelnik może najpierw rozwiązać zadanie samodzielnie, a następnie wysłać je do systemu sprawdzającego w celu zweryfikowania jego poprawności. Poniżej jest podana lista serwisów internetowych, z których zostały wybrane zadania do tej książki:

- http://acm.sgu.ru/ — Saratov State University :: Online Contester,
- http://acm.uva.es/ — Valladolid Programming Contest Site,
- http://spoj.sphere.pl/ — Sphere Online Judge.

Wszystkie zadania zostały podzielone na trzy kategorie: proste, średniej trudności oraz trudne. Poziom trudności jest rzeczą subiektywną. Podział zastosowany w książce opiera się na statystykach rozwiązań dostępnych w wyżej wymienionych serwisach. Zadania trudne charakteryzują się stosunkowo małą liczbą prób ich rozwiązania, a wśród nich niewielka część rozwiązań okazuje się poprawna. Proste zadania z kolei cechują się wysokim odsetkiem zaakceptowanych rozwiązań w stosunku do wszystkich nadesłanych programów.

Wymagania wstępne

Wszystkie przedstawione w tej książce algorytmy są zaimplementowane w języku C++. Jego znajomość jest zatem nieodzowna do zrozumiałej analizy tych algorytmów. Konieczna jest również znajomość biblioteki Standard Template Library (w skrócie STL), której dokumentację można znaleźć na stronie *http://www.sgi.com/tech/stl/*. W implementacjach algorytmów są wykorzystywane różne struktury danych oraz funkcje z tej biblioteki — najważniejsze z nich to funkcje `sort`, `swap` i `binary_search` oraz struktury danych `vector`, `map` i `priority_queue`. Oprócz powyższych elementów biblioteka STL zawiera wiele bardzo użytecznych narzędzi. Zalecane jest zapoznanie się z nią, gdyż może to w istotny sposób wpłynąć na szybkość rozwiązywania zadań algorytmicznych.

Nagłówki

Ponieważ podczas zawodów bardzo istotne jest, aby kody źródłowe implementowanych programów były jak najkrótsze, wielu zawodników stosuje pewne skróty dla najczęściej występujących w programach konstrukcji językowych. Podczas zawodów ACM ICPC, w ciągu pięciogodzinnych sesji, do rozwiązania jest 7–10 zadań, zatem opłaca się na początku konkursu przepisać najważniejsze instrukcje, a następnie używać ich we wszystkich pisanych programach. Instrukcje te są nazywane nagłówkami — zawierają one zarówno listę najczęściej dołączanych do programów bibliotek, jak i często stosowane makra. Zbiór podstawowych nagłówków pochodzących z biblioteczki algorytmicznej zespołu *Warsaw*

Predators wraz z komentarzami opisującymi ich wykorzystanie jest zamieszczony w dodatku A na listingu A.1. Same nagłówki, po usunięciu z nich komentarzy, są przedstawione na listingu A.2.

Oprócz nagłówków podstawowych istnieje również wiele innych skrótów, które okazują się pomocne podczas rozwiązywania zadań. Omawiana w tej książce biblioteczka, poza nagłówkami podstawowymi, zawiera także dodatkowe makra umieszczane tylko w tych programach, które z nich korzystają. Ich lista jest podana na listingu A.3. W prezentowanych programach będą odwołania zarówno do tych podstawowych, jak i dodatkowych nagłówków, ale ich definicje nie będą umieszczone w implementowanych programach — czytelnik powinien sam pamiętać o każdorazowym ich dopisaniu. Dzięki takiemu podejściu kody źródłowe przedstawianych w książce algorytmów są krótsze, a czytelnik nie musi za każdym razem analizować powtarzających się fragmentów kodu.

Znane są sytuacje, gdy niektórzy zawodnicy tworzyli przy użyciu makr własne, specyficzne języki programowania umożliwiające szybkie pisanie programów. Takie podejście wymaga, niestety, przepisania znacznej ilości kodu, zanim przystąpi się do właściwego rozwiązywania problemu. W przypadku konkursów organizowanych przez Internet zastosowanie takiej metody okazuje się jednak bardzo wygodne. W dodatku B znajduje się przykładowe „środowisko" programistyczne wykorzystywane w konkursie TopCoder przez Eryka Kopczyńskiego. Jest ono dość rozbudowane, dlatego też w innych konkursach (takich jak ACM ICPC) jest po prostu nieprzydatne.

Podziękowania

Opisywana w tej książce biblioteczka algorytmiczna została stworzona przy współpracy z moimi kolegami z zespołu *Warsaw Predators* — Markiem Cyganem i Marcinem Pilipczukiem. Serdecznie im dziękuję za ten wspólny trud. Chciałbym również podziękować Tomaszowi Idziaszkowi, którego kilka „chwytów programistycznych" znalazło się w naszej biblioteczce, oraz Erykowi Kopczyńskiemu, który zgodził się na opublikowanie zbioru swoich makr używanych w konkursie TopCoder.

Jestem niezmiernie wdzięczny Tomaszowi Czajce, Andrzejowi Gąsienicy-Samkowi, Tomaszowi Malesińskiemu, Krzysztofowi Onakowi i Marcinowi Stefaniakowi za wyrażenie zgody na publikację ich „dobrych rad", których udzielili swoim młodszym kolegom z Uniwersytetu Warszawskiego, biorącym udział w konkursie ACM ICPC. Zostały one zebrane w dodatku C tej książki.

Ważnym wyzwaniem podczas gromadzenia materiałów do książki było wybieranie odpowiednich zadań z różnych serwisów internetowych. Liczba dostępnych zadań jest ogromna (w momencie pisania tego tekstu serwis Valladolid Programming Contest Site zawierał ponad dwa tysiące zadań). Chciałbym podziękować Jakubowi Radoszewskiemu za udzielenie cennych wskazówek dotyczących wyboru zadań.

Szczególne podziękowania należą się profesorowi Krzysztofowi Diksowi — naszemu wykładowcy i wielkiemu przyjacielowi, który wytrwale opiekował się nami i stwarzał warunki pozwalające na efektywne przygotowania naszych drużyn do zawodów. Niniejsza książka powstała pod jego życzliwym nadzorem.