W PROSTOCIE TKWI SIŁA

Algorytmy

dla bystrzaków

Zestaw algorytmy z ich zastosowaniami

Zdobądź umiejętności posługiwania się algorytmami

Naucz się wykorzystywać Pythona do testowania algorytmów

John Paul Mueller
Luca Massaron

septem
septem.pl
Helion

Tytuł oryginału: Algorithms For Dummies

Tłumaczenie: Radosław Meryk

ISBN: 978-83-283-6076-1

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://dlabystrzakow.pl/user/opinie/algoby
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:
ftp://ftp.helion.pl/przyklady/algoby.zip

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 231 22 19, 32 230 98 63
e-mail: *dlabystrzakow@dlabystrzakow.pl*
WWW: *http://dlabystrzakow.pl*

Printed in Poland.

Spis treści

W TYM ROZDZIALE:

- Co rozumiemy przez algorytm?
- Wykorzystywanie komputerów do dostarczania rozwiązań za pomocą algorytmów
- Ustalenie różnic pomiędzy problemami a rozwiązaniami
- Operowanie na danych w celu znalezienia rozwiązań

Rozdział 1

Wprowadzenie do algorytmów

Jeśli jesteś podobny do większości ludzi, zapewne poczujesz się zagubiony, otwierając tę książkę i rozpoczynając przygodę z algorytmami. W tekstach dotyczących algorytmów przeważnie bowiem nie wspomina się, czym są algorytmy, a tym bardziej po co się ich używa. W większości tekstów zakłada się, że już czytelnik coś wie o algorytmach i czyta o nich po to, aby wzbogacić swoją wiedzę. Co ciekawe, niektóre książki wprowadzają mylącą definicję algorytmu, która w zasadzie go nie definiuje, a czasem nawet utożsamia z jakąś inną formą abstrakcyjnego, liczbowego lub symbolicznego wyrażenia.

Pierwsza część tego rozdziału ma za zadanie pomóc Ci zrozumieć, co to jest algorytm i dlaczego znajomość algorytmów jest przydatna. Algorytmy nie są wiedzą tajemną — używa się ich wszędzie. Prawdopodobnie korzystasz z nich od lat, wcale o tym nie wiedząc. Algorytmy są kręgosłupem, który wspiera i reguluje to, co jest ważne w coraz bardziej złożonym i technicznym społeczeństwie.

W tym rozdziale przybliżamy także, jak wykorzystywać komputery do tworzenia rozwiązań problemów przy użyciu algorytmów, jak rozróżniać problemy od rozwiązań oraz jak operować danymi, aby znaleźć rozwiązanie. Celem tego rozdziału

jest pomoc w odróżnieniu algorytmów od innych zadań, które ludzie wykonują i które mylą z algorytmami. Krótko mówiąc, w tym rozdziale odkryjesz, do czego są Ci potrzebne algorytmy i jak je zastosować do danych.

Co to jest algorytm?

Chociaż ludzie stosowali algorytmy ręcznie przez tysiące lat, takie podejście pochłania ogrom czasu i wymaga wielu obliczeń numerycznych, w zależności od złożoności problemu, który chcemy rozwiązać. Algorytmy dotyczą znajdowania rozwiązań — im szybciej i łatwiej, tym lepiej. Istnieje ogromna luka między algorytmami matematycznymi historycznie stworzonymi przez geniuszy swoich czasów, takich jak Euklides, Newton czy Gauss, a nowoczesnymi algorytmami opracowanymi na uniwersytetach i w prywatnych laboratoriach badawczo-rozwojowych. Głównym powodem tej luki jest wykorzystanie komputerów. Ich stosowanie do rozwiązywania problemów za pomocą odpowiednich algorytmów pozwala znacznie szybciej wykonywać zadania. Właśnie dlatego — od czasów pojawienia się potężnych systemów komputerowych — rozwój nowych algorytmów postępuje tak szybko. Prawdopodobnie zauważyłeś, że współcześnie pojawia się coraz więcej rozwiązań problemów. Częściowo wynika to stąd, że moc obliczeniowa komputerów jest zarówno tania, jak i stale rośnie. Komputery (czasami w postaci specjalnego sprzętu) stają się wszechobecne właśnie z powodu ich zdolności do rozwiązywania problemów przy użyciu algorytmów.

Podczas pracy z algorytmami bierze się pod uwagę dane wejściowe, pożądane dane wynikowe i proces (ciąg działań) prowadzący do uzyskania pożądanego wyniku na podstawie danych wejściowych. Możesz jednak źle zrozumieć terminologię i niewłaściwie postrzegać algorytmy, ponieważ nigdy nie zastanawiałeś się nad tym, jak w rzeczywistości działają. W trzecim podrozdziale tego rozdziału omówiliśmy algorytmy w świecie rzeczywistym, to znaczy dokonaliśmy przeglądu terminologii stosowanej w celu zrozumienia algorytmów i zaprezentowaliśmy algorytmy w sposób, który dowodzi, że rzeczywisty świat jest często daleki od doskonałości. Umiejętność realistycznego opisania algorytmu pozwala także złagodzić oczekiwania wobec algorytmów i odzwierciedlić rzeczywisty stan tego, co faktycznie można za ich pomocą osiągnąć.

Ta książka prezentuje algorytmy na wiele sposobów. Ponieważ jednak dostarcza opisu sposobu, w jaki algorytmy się zmieniają i wzbogacają życie ludzi, koncentruje się na algorytmach wykorzystywanych do przetwarzania danych z użyciem komputerów. To one dostarczają wymaganej mocy obliczeniowej. Mając to na uwadze, algorytmy, z którymi pracujesz w tej książce, wymagają wprowadzania danych w określonej formie, co czasami oznacza modyfikowanie danych w celu ich dopasowania do wymagań algorytmu. Przetwarzanie danych nie zmienia ich zawartości. Zmienia jedynie prezentację i formę danych. Dzięki temu algorytmy mogą pomóc Ci zauważyć nowe wzorce — takie, które wcześniej nie były widoczne (choć występowały w danych przez cały czas).

Źródła informacji o algorytmach często przedstawiają je w sposób, który jest mylący — zbyt wyrafinowany lub wręcz niepoprawny. W tej książce użyto następujących definicji dla terminów, które ludzie często mylą z algorytmami:

- **Równanie** — liczby i symbole, które w całości są równe określonej wartości. Równanie zawsze zawiera znak równości, dzięki czemu wiemy, że liczby i symbole reprezentują określoną wartość po drugiej stronie znaku równości. Równania na ogół zawierają zmienne informacje zaprezentowane w symbolicznej formie, choć nie muszą używać zmiennych.
- **Wzór (formuła)** — połączenie liczb i symboli stosowane do wyrażenia informacji lub pojęć. Wzory zazwyczaj przedstawiają pojęcia matematyczne lub logiczne, takie jak definiowanie największego wspólnego dzielnika (NWD) dwóch liczb całkowitych (aby dowiedzieć się, co to jest, można obejrzeć film dostępny pod adresem *https://www.khanacademy.org/math/in-sixth-grade-math/playing--numbers/highest-common-factor/v/greatest-common-divisor*). Ogólnie rzecz biorąc, wzory pokazują związek pomiędzy co najmniej dwiema zmiennymi. Większość osób uważa wzory za specjalny rodzaj równań.
- **Algorytm** — sekwencja działań zmierzających do rozwiązania problemu. Sekwencja przedstawia unikatową metodę podejścia do problemu poprzez dostarczenie konkretnego rozwiązania. Algorytm nie musi reprezentować pojęć matematycznych lub logicznych. Algorytmy zaprezentowane w tej książce często należą jednak do tej kategorii, ponieważ ludzie przeważnie stosują algorytmy w taki sposób. Niektóre specjalne wzory, na przykład równania kwadratowe, to także algorytmy. Aby proces reprezentował algorytm, musi być:
  - **Skończony** — algorytm musi prowadzić do rozwiązania problemu. W tej książce omawiamy problemy ze znanymi rozwiązaniami, dzięki czemu możemy ocenić, czy algorytm rozwiązuje problem poprawnie.
  - **Ściśle zdefiniowane** — sekwencja kroków musi być precyzyjna i musi przedstawiać kroki w sposób zrozumiały. Ponieważ w wykorzystanie algorytmu są zaangażowane komputery, muszą one być w stanie zrozumieć kroki zmierzające do stworzenia użytecznego algorytmu.
  - **Skuteczny** — algorytm musi uwzględniać wszystkie przypadki problemu, dla którego ktoś go zdefiniował. Algorytm powinien zawsze rozwiązywać problem, do którego rozwiązania jest przeznaczony. Nawet jeśli przewidujemy jakieś awarie, częstość awarii powinna być niska. Awarie występuje tylko w sytuacjach, które są dopuszczalne dla zamierzonego użycia algorytmu.

W dalszych podrozdziałach staramy się opisać naturę algorytmów z uwzględnieniem powyższych definicji. Celem nie jest zapewnienie precyzyjnej definicji algorytmów, ale raczej pomoc w zrozumieniu ich istoty, dzięki czemu będziesz w stanie rozwijać własne rozumienie tego, czym są algorytmy i dlaczego są tak ważne.

Zastosowania algorytmów

Algorytm zawsze prezentuje sekwencję kroków, choć niekoniecznie wykonuje te kroki w celu rozwiązania matematycznego równania. Zakres algorytmów jest niewiarygodnie duży. Istnieją algorytmy, które rozwiązują problemy w nauce, medycynie, finansach, produkcji przemysłowej, a także logistyce i komunikacji. Algorytmy zapewniają wsparcie dla wszystkich aspektów codziennego życia. Za każdym razem, gdy sekwencja działań, które do czegoś prowadzą, jest skończona, dobrze zdefiniowana i skuteczna, możemy postrzegać ją jako algorytm. Na przykład możesz przekształcić w algorytm nawet coś tak trywialnego i prostego jak przygotowanie tostu. Procedura przygotowywania tostów często pojawia się na lekcjach informatyki. Taki opis można znaleźć na stronie *http://brianaspinall.com/now-thats--how-you-make-toast-using-computer-algorithms/*.

Niestety algorytm zaprezentowany na tej stronie jest wadliwy. Instruktor nigdy nie wyjmuje tostu z opakowania i nigdy nie wkłada go do tostera, w wyniku czego powstaje uszkodzona kromka wciąż w opakowaniu w niedziałającym tosterze (szczegółowe informacje można znaleźć na stronie *http://blog.johnmuellerbooks.com/2013/03/04/procedures-in-technicalwriting/*). Mimo że koncepcja jest poprawna, to aby algorytm był skończony i skuteczny, wymaga pewnych drobnych, ale koniecznych zmian.

Jednym z najczęstszych zastosowań algorytmów jest rozwiązywanie wzorów. Na przykład NWD dwóch liczb całkowitych można obliczyć ręcznie. W tym celu należy podać wszystkie dzielniki obu liczb całkowitych, a następnie wybrać największy dzielnik, który jest dla nich wspólny. Na przykład NWD(20, 25) wynosi 5, ponieważ 5 jest największą liczbą będącą dzielnikiem zarówno liczby 20, jak i 25. Ręczne obliczanie każdego NWD (co w istocie jest rodzajem algorytmu) to jednak proces czasochłonny i stwarzający okazje do popełnienia błędów. Z tych powodów grecki matematyk Euklides (*https://pl.wikipedia.org/wiki/Euklides*) stworzył algorytm do wykonania tego zadania. Z metodą Euklidesa można zapoznać się na stronie *https://www.khanacademy.org/computing/computerscience/cryptography/modarithmetic/a/the-euclidean-algorithm*.

Pojedynczy wzór, który jest prezentacją symboli i liczb używanych do wyrażenia informacji lub pojęć, może mieć wiele rozwiązań. Każde z nich jest algorytmem. W przypadku NWD innym popularnym algorytmem jest algorytm Lehmera (zobacz *https://www.imsc.res.in/~kapil/crypto/notes/node11.html* i *https://en.wikipedia.org/wiki/Lehmer%27s_GCD_algorithm*). Ponieważ każdy wzór można rozwiązać na wiele sposobów, często spędzamy dużo czasu, porównując algorytmy — aby znaleźć taki algorytm, który najlepiej sprawdza się w określonej sytuacji (zobacz porównanie algorytmów Euklidesa i Lehmera pod adresem *http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.31.693&rep=rep1&type=pdf*).

Ponieważ nasze społeczeństwo i towarzysząca mu technologia nabierają coraz większego tempa, potrzebujemy algorytmów, które potrafią to tempo utrzymać. W naszych czasach możliwe stały się takie osiągnięcia naukowe jak sekwencjo-

nowanie ludzkiego genomu dzięki temu, że naukowcy znaleźli algorytmy działające wystarczająco szybko, aby wykonać to zadanie. Zmierzenie, który algorytm jest lepszy w danej sytuacji lub w przeciętnej sytuacji użytkownika, to naprawdę poważny problem i temat do dyskusji wśród informatyków.

W informatyce ten sam algorytm może mieć wiele postaci. Na przykład algorytm Euklidesa można zaprezentować zarówno w sposób rekurencyjny, jak i iteracyjny, co objaśniono na stronie *http://cs.stackexchange.com/questions/1447/what-is-most--efficient-for-gcd*. Ogólnie rzecz biorąc, algorytmy przedstawiają metodę rozwiązywania wzorów. Błędem byłoby jednak stwierdzenie, że istnieje tylko jeden akceptowalny algorytm dla danego wzoru lub że istnieje tylko jedna akceptowalna postać algorytmu. Wykorzystywanie algorytmów do rozwiązywania różnego rodzaju problemów ma długą historię — to nie jest coś, co właśnie się wydarzyło.

Nawet jeśli ograniczysz swoje poszukiwania do informatyki, inżynierii danych, sztucznej inteligencji i innych dziedzin technicznych, znajdziesz wiele rodzajów algorytmów — zbyt wiele jak na jedną książkę. Na przykład książka *The Art of Computer Programming* Donalda E. Knutha (Addison-Wesley) ma 3168 stron w czterech tomach (zobacz *http://www.amazon.com/exec/obidos/ASIN/0321751043/datacservipof-20/*) i mimo to nie jest w stanie wyczerpać tematu (autor zamierzał napisać więcej tomów). Oto kilka interesujących zastosowań algorytmów:

- **Wyszukiwanie** — znajdowanie informacji lub sprawdzanie, czy informacje, które widzimy, są potrzebne, to zadanie o podstawowym znaczeniu. Bez takich mechanizmów nie byłoby możliwe wykonanie wielu zadań w trybie online, takich jak na przykład znalezienie strony internetowej oferującej idealny dzbanek do kawy do Twojego biura.
- **Sortowanie** — ustalenie kolejności, w jakiej wykorzystujemy informacje, jest bardzo ważne. Dla wielu osób natłok informacji to duży problem, a uporządkowanie ich jest jednym ze sposobów na zmniejszenie napływu danych. Prawdopodobnie będąc dzieckiem, nauczyłeś się, że kiedy utrzymywałeś porządek w zabawkach, łatwiej było Ci je znaleźć i pobawić się tą zabawką, która Cię akurat interesowała, niż wtedy, gdy zabawki były porozrzucane po całym domu. Wyobraź sobie, że wchodzisz do serwisu Amazon i dowiadujesz się, że ma sprzedaży ponad tysiąc dzbanków do kawy i nie możesz posortować ich według ceny lub najbardziej pozytywnej opinii. Co więcej, wiele złożonych algorytmów wymaga danych w odpowiedniej kolejności do niezawodnego działania, dlatego sortowanie jest ważnym wymogiem dla rozwiązania wielu problemów.
- **Przekształcanie** — konwersja jednego rodzaju danych na inny rodzaj danych ma kluczowe znaczenie dla ich efektywnego zrozumienia i wykorzystania. Na przykład możesz dobrze rozumieć wagę w skali brytyjskiej, ale we wszystkich Twoich źródłach jest stosowany system metryczny. Konwersja pomiędzy tymi dwoma systemami pomaga je zrozumieć. Podobnie szybka transformata Fouriera (FFT) przekształca sygnały pomiędzy domeną czasu a domeną częstotliwości, dzięki czemu możliwe staje się działanie takich urządzeń jak router wi-fi.

- **Szeregowanie** — doprowadzenie do sprawiedliwego podziału zasobów przez wszystkie zainteresowane podmioty to kolejny obszar, w którym algorytmy znacząco ujawniają swoją obecność. Na przykład synchronizacja świateł na skrzyżowaniach nie sprowadza się już tylko do odliczania sekund pomiędzy zmianami świateł. Nowoczesne urządzenia uwzględniają wiele czynników, takich jak pora dnia, warunki pogodowe czy natężenie ruchu. Szeregowanie ma jednak wiele postaci. Na przykład zastanów się, w jaki sposób komputer wykonuje jednocześnie wiele zadań. Bez algorytmu szeregowania system operacyjny mógłby pobrać wszystkie dostępne zasoby i uniemożliwić aplikacji wykonanie pożytecznej pracy.
- **Analiza grafów** — wybór najkrótszej drogi pomiędzy dwoma punktami znajduje wiele różnych zastosowań. Na przykład bez tego konkretnego algorytmu nie mogłaby funkcjonować nawigacja GPS, ponieważ nie potrafiłaby skierować Cię po ulicach miasta przy użyciu najkrótszej trasy z punktu A do punktu B.
- **Kryptografia** — utrzymywanie bezpieczeństwa danych to nieustanna walka z hakerami stale atakującymi źródła danych. Algorytmy umożliwiają analizę danych, ich transformację do określonej postaci, a następnie powrót do postaci pierwotnej.
- **Generowanie liczb pseudolosowych** — wyobraź sobie, że grasz w gry, które nigdy się nie zmieniają. Przy każdej rozgrywce zaczynasz w tym samym miejscu i w ten sam sposób wykonujesz te same kroki. Bez możliwości generowania pozornie losowych liczb realizacja wielu komputerowych zadań stałaby się niemożliwa.

Powyższa lista to bardzo krótki przegląd zastosowań algorytmów. Algorytmów używa się do rozwiązywania wielu różnych zadań i na wiele różnych sposobów. Stale powstają nowe algorytmy do rozwiązania zarówno istniejących, jak i nowych problemów. Najważniejszą kwestią, którą należy uwzględnić w pracy z algorytmami, jest to, że przy danym wejściu należy oczekiwać określonego wyjścia. Problemy drugorzędne to ilość zasobów, których algorytm wymaga do wykonania swojego zadania, oraz czas potrzebny do jego wykonania. W zależności od rodzaju problemu i rodzaju zastosowanego algorytmu może być również konieczne uwzględnienie kwestii dokładności i spójności.

Algorytmy są wszędzie

W poprzednim podrozdziale nieprzypadkowo wspomnieliśmy algorytm przygotowywania tostów. Z jakiegoś powodu robienie tostów jest jednym z najpopularniejszych algorytmów, jakie kiedykolwiek powstały. Wielu uczniów szkół ponadgimnazjalnych pisze odpowiedniki algorytmu przygotowywania tostów na długo przed tym, zanim rozwiążą najbardziej podstawowe zadanie z matematyki. Nietrudno sobie wyobrazić, jak wiele odmian algorytmu tostowego istnieje i jakie

są wyniki każdego z nich. Prawdopodobnie wyniki są różne w zależności od osoby i poziomu kreatywności. Krótko mówiąc, algorytmy pojawiają się w dużej różnorodności, często w nieoczekiwanych miejscach.

Algorytmy wykorzystuje się we wszystkich zadaniach wykonywanych na komputerach. Niektóre algorytmy są elementami sprzętu komputerowego (są wbudowane, dlatego często mówimy o wbudowanych mikroprocesorach). Sam proces uruchamiania komputera obejmuje użycie algorytmu. Algorytmy są wykorzystywane również w systemach operacyjnych, aplikacjach i każdym innym oprogramowaniu. Nawet użytkownicy polegają na algorytmach. Skrypty pomagają użytkownikom wykonywać zadania w określony sposób, ale te same kroki mogą mieć postać pisemnych instrukcji lub tworzyć strategię organizacji.

Codzienne czynności często zmieniają się w algorytmy. Pomyśl o tym, jak spędzasz dzień. Zazwyczaj, jak większość ludzi, wykonujesz każdego dnia te same zadania w tej samej kolejności. W związku z tym Twój dzień staje się algorytmem, który rozwiązuje problem wygodnego życia przy użyciu możliwie jak najmniejszej ilości energii. W końcu właśnie taki cel spełnia rutyna — czyni nas skutecznymi.

Na algorytmach często bazują procedury awaryjne. Instrukcję postępowania w sytuacjach awaryjnych wyjmujesz z kieszeni znajdującego się przed Tobą fotela w samolocie. Ma ona postać serii piktogramów pokazujących, jak otworzyć drzwi awaryjne i wysunąć awaryjną zjeżdżalnię. Niekiedy algorytmy nie zawierają nawet słów. Same rysunki ilustrują procedurę wymaganą do wykonania zadania i rozwiązania problemu awaryjnego wysiadania z samolotu. W tej książce dla każdego algorytmu zobaczymy te same trzy elementy:

1. Opisz problem.
2. Utwórz ciąg (dobrze zdefiniowanych) kroków prowadzących do rozwiązania problemu.
3. Wykonaj te kroki, aby uzyskać pożądany rezultat (skończony i skuteczny).

Stosowanie komputerów do rozwiązywania problemów

Słowo *komputer* brzmi dość technicznie i może być nieco przytłaczające dla niektórych osób. Współczesny człowiek tkwi jednak w komputerach „po uszy” (a być może nawet głębiej). Przez większość czasu masz przy sobie przynajmniej jeden komputer — Twój smartfon. Jeśli masz jakieś specjalne urządzenie, na przykład rozrusznik serca, ono także zawiera komputer. Twój telewizor zawiera co najmniej jeden komputer. W komputery są również wyposażone „inteligentne” urządzenia AGD. Samochód może zawierać nawet 30 komputerów — mają one

postać wbudowanych mikroprocesorów sterujących zużyciem paliwa, emisją spalin, działaniem skrzyni biegów i układu kierowniczego oraz stabilnością (zobacz artykuł z „New York Timesa" dostępny na stronie *http://www.nytimes.com/2010/02/05/technology/05electronics.html*) — i więcej wierszy kodu niż odrzutowy myśliwiec. Pojawiające się na rynku motoryzacyjnym samochody autonomiczne będą jeszcze bardziej wymagały wbudowanych mikroprocesorów i algorytmów o jeszcze większej złożoności. Komputer służy do szybkiego rozwiązywania problemów i z mniejszym wysiłkiem niż w przypadku rozwiązywania ich ręcznie. Dlatego nie powinno Cię dziwić, że w tej książce jeszcze bardziej wykorzystujemy komputery po to, aby lepiej zrozumieć algorytmy.

Komputery różnią się na wiele sposobów. Komputer w zegarku jest dość mały — ten, który znajduje się na Twoim biurku, jest stosunkowo duży. Superkomputery są ogromne i składają się z wielu mniejszych komputerów — wszystkie one współpracują ze sobą podczas rozwiązywania złożonych problemów, takich jak opracowywanie prognozy pogody na jutro. Najbardziej złożone algorytmy bazują na specjalnych funkcjonalnościach komputerów. Za ich pomocą uzyskujemy rozwiązania problemów, do których rozwiązywania te funkcjonalności zaprojektowano. To prawda, że można zużyć mniejsze zasoby, aby wykonać zadanie, ale kosztem uzyskania odpowiedzi po znacznie dłuższym czasie lub uzyskania odpowiedzi, która nie jest wystarczająco dokładna, by była użytecznym rozwiązaniem. W niektórych przypadkach na odpowiedź czekamy tak długo, że w momencie jej uzyskania nie jest ona już ważna. Biorąc pod uwagę zarówno szybkość, jak i dokładność, w poniższych punktach omówiliśmy kilka specjalnych własności komputera, które mogą mieć wpływ na algorytmy.

Wykorzystanie nowoczesnych procesorów i procesorów graficznych

Procesory ogólnego przeznaczenia — CPU — powstały jako mechanizmy do rozwiązywania problemów z wykorzystaniem algorytmów. Jednak z ich uniwersalnego charakteru wynika również to, że procesory mogą wykonywać wiele innych zadań, takich jak na przykład przenoszenie danych lub interakcje z urządzeniami zewnętrznymi. Procesory ogólnego przeznaczenia mają wiele zalet, co oznacza, że mogą wykonywać kroki potrzebne do realizacji algorytmu, choć niekoniecznie szybko. Właściciele wczesnych procesorów ogólnego przeznaczenia mogli dodatkowo instalować w swoich systemach koprocesory matematyczne (specjalne układy służące do wykonywania działań arytmetycznych). Dzięki temu mogli zyskać większą szybkość działania (więcej informacji można znaleźć na stronie *http://www.computerhope.com/jargon/m/mathcopr.htm*). Obecnie procesory ogólnego przeznaczenia mają wbudowany koprocesor matematyczny, zatem kiedy kupujesz procesor Intel i7, w rzeczywistości otrzymujesz wiele procesorów w jednym pakiecie.

Co ciekawe, Intel nadal wprowadza na rynek specjalne dodatki do procesorów, takie jak procesor Xeon Phi używany z układami Xeon (więcej informacji można znaleźć na stronach *http://www.intel.com/content/www/us/en/processors/xeon/xeon-phi-detail.*
↪html i *https://en.wiki2.org/wiki/Intel_Xeon_Phi*). Układ Xeon Phi wraz z układem Xeon służy do wykonywania zadań wymagających dużej mocy obliczeniowej, takich jak uczenie maszynowe (szczegółowe informacje o tym, jak uczenie maszynowe wykorzystuje algorytmy do ustalenia sposobu wykonywania różnych zadań wspomagających użycie danych oraz do przewidywania nieznanych informacji i porządkowania ich tak, by na ich podstawie wyciągać sensowne wnioski, można znaleźć w książce Johna Muellera i Luki Massarona *Machine Learning For Dummies*).

Być może zastanawiasz się, dlaczego w tym punkcie wspominamy o procesorach graficznych (GPU). W końcu GPU mają pobierać dane, w specjalny sposób je przetwarzać, a następnie wyświetlać na ekranie estetyczny obraz. Każdy sprzęt komputerowy może służyć do więcej niż jednego celu. Okazuje się, że procesory GPU są szczególnie wydajne w transformacjach danych, co w wielu przypadkach jest jednym z najważniejszych zadań przy rozwiązywaniu algorytmów. GPU to procesory specjalnego przeznaczenia, które mają właściwości umożliwiające szybsze wykonywanie algorytmów. Nie powinno nikogo dziwić, że osoby, które opracowują algorytmy, spędzają dużo czasu na myśleniu. W związku z tym często znajdują sposoby rozwiązywania problemów za pomocą nietradycyjnych metod.

Chodzi o to, że CPU i GPU to najczęściej używane układy do wykonywania zadań związanych z algorytmami. Pierwsze całkiem dobrze wykonują zadania ogólnego przeznaczenia, natomiast drugie specjalizują się w zapewnieniu obsługi zadań wymagających intensywnych obliczeń matematycznych — zwłaszcza tych, które wymagają transformacji danych. Korzystanie z wielu rdzeni umożliwia przetwarzanie równoległe (wykonywanie więcej niż jednego kroku algorytmu naraz). Dodanie wielu układów zwiększa liczbę dostępnych rdzeni. Posiadanie większej liczby rdzeni zwiększa szybkość, ale z powodu wielu czynników zysk szybkości nie jest liniowy: użycie w systemie dwóch układów i7 nie spowoduje podwojenia szybkości w porównaniu z systemem wyposażonym tylko w jeden układ i7.

Wykorzystanie układów specjalnych

Koprocesor matematyczny i procesor graficzny to dwa przykłady popularnych układów specjalnych — takich, które nie są używane do wykonywania zadań ogólnych, na przykład ładowania systemu. Algorytmy często jednak wymagają użycia nietypowych układów specjalnych do rozwiązywania problemów. Niniejsza książka nie jest o sprzęcie. Warto jednak poświęcić trochę czasu, aby się rozejrzeć i zauważyć wiele interesujących układów, takich jak nowe sztuczne neurony, nad którymi pracuje IBM (zobacz historię na stronie *http://www.computerworld.com/article/3103294/computer-processors/ibm-creates-artificial-neurons-from-phase-change--memory-for-cognitive-computing.html*). Wyobraź sobie algorytmiczne przetwarzanie z wykorzystaniem pamięci, która symuluje ludzki mózg. Byłoby to interesujące środowisko do wykonywania zadań, których wykonanie bez takiego rodzaju pamięci nie byłoby dzisiaj możliwe.

Sieci neuronowe, technologia wykorzystywana do symulacji myśli człowieka i umożliwiająca stosowanie technik uczenia głębokiego w scenariuszach uczenia maszynowego, korzystają obecnie z wyspecjalizowanych układów scalonych, takich jak Tesla P100 firmy NVidia (aby poznać szczegóły, zobacz artykuł na stronie *https://www.technologyreview.com/s/601195/a-2-billionchip-to-accelerate-artificial-intelligence/*). Tego rodzaju układy nie tylko wykonują niezwykle szybkie przetwarzanie algorytmiczne, ale uczą się, jak wykonywać zadania, dzięki czemu są szybsze przy każdej kolejnej iteracji. Komputery uczące się w końcu zaczną być instalowane w robotach, które będą mogły same się poruszać, podobnie jak roboty zaprezentowane w filmie *Ja, robot* (jeden z takich robotów został opisany na stronie *http://www.cbsnews.com/news/this-creepy-robot-is-powered-by-a-neuralnetwork/*). Istnieją również specjalne układy, które wykonują takie zadania jak rozpoznawanie obrazów (szczegółowe informacje można znaleźć na stronie *https://www.technolo--gyreview.com/s/537211/a-better-way-tobuild-brain-inspired-chips/*).

Niezależnie od tego, jak działają procesory specjalizowane, w końcu zostaną one wykorzystane do wykonywania wszystkich rodzajów algorytmów, co będzie miało konsekwencje w świecie rzeczywistym. Wiele z tych rzeczywistych aplikacji w stosunkowo prostej formie można znaleźć już dziś. Na przykład wyobraź sobie zadania, które musiałby rozwiązać robot do pizzy — zmienne, które musiałby przetwarzać w czasie rzeczywistym. Takie roboty już istnieją (jest to tylko jeden przykład wielu robotów przemysłowych wykorzystywanych do produkcji dóbr materialnych przy użyciu algorytmów). Można założyć, że jego działanie bazuje na algorytmach, które opisują co robić, a także na specjalnych układach, dzięki którym zadania są wykonywane szybko (zobacz artykuł na stronie *http://www.bloomberg.com/news/articles/2016-06-24/inside-silicon-valley-s-robot-pizzeria*).

Być może kiedyś powstanie możliwość wykorzystania w roli procesora ludzkiego umysłu i wyprowadzenia informacji przez specjalny interfejs. Istnieją firmy, które eksperymentują z umieszczaniem procesorów bezpośrednio w ludzkim mózgu, aby zwiększyć jego zdolność do przetwarzania informacji (zobacz artykuł na stronie ,*https://www.washingtonpost.com/news/the-switch/wp/2016/08/15/putting--a-computer-in-your-brainis-no-longer-science-fiction/*). Wyobraź sobie system, w którym ludzie mogą stosować algorytmy z szybkością komputerów, ale z kreatywnym potencjałem „co jeśli" charakterystycznym dla ludzi.

Wykorzystanie sieci

Jeśli Twoje fundusze nie są nieograniczone, skuteczne wykorzystanie niektórych algorytmów może być niemożliwe, nawet przy użyciu specjalistycznych układów. W takim przypadku istnieje możliwość połączenia komputerów w sieć. Dzięki wykorzystaniu specjalnego oprogramowania jeden komputer — tzw. *master* — może używać procesorów wszystkich komputerów podrzędnych z uruchomionym agentem (rodzajem aplikacji uruchomionej w tle, która udostępnia procesor). Korzystając z tego podejścia, można rozwiązać złożone problemy poprzez przekazanie części problemu do wielu komputerów podrzędnych. W miarę jak każdy komputer w sieci rozwiązuje swoją część problemu, przesyła wyniki z po-

wrotem do mastera, który scala elementy w celu stworzenia skonsolidowanej odpowiedzi. Taka technika nazywa się *przetwarzaniem w klastrze* (ang. *cluster computing*).

Nie jest to bynajmniej temat science fiction. Obecnie ludzie używają technik obliczeń w klastrze na wiele interesujących sposobów. Na przykład w artykule na stronie *http://www.zdnet.com/article/build-your-own-supercomputer-out-ofraspberry-pi-boards/* zawarto szczegółowe informacje o tym, jak zbudować własny superkomputer poprzez połączenie w klaster wielu mikrokontrolerów Raspberry Pi (zobacz stronę *https://www.raspberrypi.org/*).

Popularna jest także technologia *przetwarzania rozproszonego* — innej wersji obliczeń klastrowych (ale z luźniejszą organizacją). Listę projektów, w których zastosowano przetwarzanie rozproszone, można znaleźć na stronie *http://www.distributioncomputing.info/projects.html*. Wśród nich jest kilka ważnych przedsięwzięć, takich jak wyszukiwanie inteligencji pozaziemskiej (projekt SETI). Możesz także przekazać swoją moc obliczeniową komputera do pracy nad lekiem na raka. Lista potencjalnych projektów jest niezwykła.

Sieci umożliwiają również dostęp do mocy obliczeniowej innych komputerów w formie niezwiązanej. Na przykład mechanizmy do korzystania z komputerów w celu wykonywania własnej pracy zapewnia usługa Amazon Web Services (AWS), a także inni dostawcy. Dzięki połączeniu sieciowemu użytkownicy mogą odnosić wrażenie, że komputery zdalne należą do ich własnej sieci. Chodzi o wykorzystanie sieci na różne sposoby do tworzenia połączeń między komputerami w celu rozwiązywania różnych algorytmów, które byłyby zbyt skomplikowane, aby je rozwiązać za pomocą tylko jednego systemu.

Wykorzystywanie dostępnych danych

Część zadań związanych ze stosowaniem algorytmu nie ma nic wspólnego z mocą obliczeniową, twórczym myśleniem lub z czymkolwiek o fizycznym charakterze. Aby stworzyć rozwiązanie większości problemów, potrzebne są również dane, na których można oprzeć swoje wnioski. Na przykład w algorytmie do przygotowywania tostów, zanim faktycznie rozwiążesz problem robienia tostów, musisz wiedzieć, czy jest dostępny chleb, toster, prąd elektryczny do zasilania tostera i tak dalej. Dane te stają się ważne, ponieważ nie można ukończyć algorytmu, gdy brakuje choćby jednego elementu wymaganego rozwiązania. Oczywiście mogą być także potrzebne dodatkowe dane wejściowe. Na przykład osoba, która robi tost, może nie lubić żytniego pieczywa. Jeśli tak jest, a mamy tylko żytni chleb do dyspozycji, to pomimo dostępności chleba nadal nie uzyskamy pomyślnego wyniku.

Dane pochodzą z różnych źródeł i mogą mieć bardzo różne formaty. Możesz przesyłać dane ze źródła takiego jak monitor czasu rzeczywistego, strumieniowo, sięgać do publicznego źródła danych, korzystać z prywatnych danych w bazie danych, pobierać dane ze stron internetowych za pomocą techniki *scrapingu* lub uzyskiwać je niezliczonymi sposobami — zbyt wieloma, by je tutaj wymienić.

Dane mogą być statyczne (niezmienne) lub dynamiczne (stale się zmieniające). Może się okazać, że dane są kompletne lub brakuje im kilku elementów. Dane mogą mieć nieodpowiednią formę (na przykład waga jest wyrażona w jednostkach skali brytyjskiej, a są potrzebne jednostki metryczne). Dane mogą mieć postać tabelaryczną w sytuacji, gdy są potrzebne w innej formie. Mogą być nieuporządkowane (na przykład w bazie danych NoSQL lub po prostu w wielu różnych plikach danych), a my potrzebujemy formalnego formatowania właściwego dla relacyjnej bazy danych. Krótko mówiąc, aby rozwiązywać problemy za pomocą algorytmu, musisz znać bardzo wiele różnych danych.

Ponieważ dane występują w bardzo wielu formach i trzeba z nimi pracować na wiele sposobów, w tej książce przywiązuje się do danych olbrzymią wagę. Począwszy od rozdziału 6., dowiesz się, jaką rolę odgrywa struktura danych. W rozdziale 7. pokażemy, w jaki sposób przeszukiwać dane, aby znaleźć to, co jest potrzebne. W rozdziałach od 12. do 14. omawiamy sposoby pracy z dużymi zbiorami danych. Jednak w każdym rozdziale książki można znaleźć pewne informacje specyficzne dla danych. Bez danych algorytm nie jest w stanie rozwiązać żadnych problemów.

Odróżnianie problemów od rozwiązań

W tej książce omawiamy dwie części algorytmicznego widoku rzeczywistego świata. Z jednej strony mamy *problemy* do rozwiązania. Problem może opisywać pożądany wynik algorytmu lub przeszkodę, którą należy pokonać, aby uzyskać pożądany wynik. *Rozwiązania* to metody lub kroki stosowane w celu rozwiązania problemów. Rozwiązanie może dotyczyć tylko jednego kroku lub wielu kroków w algorytmie. Rozwiązanie to wynik algorytmu, czyli odpowiedź na jego ostatni krok. Niektóre ważne aspekty problemów i rozwiązań zrozumiesz dzięki lekturze zamieszczonych poniżej punktów.

Poprawność a skuteczność

Korzystanie z algorytmów polega na uzyskaniu akceptowalnej odpowiedzi. Szukamy akceptowalnej odpowiedzi dlatego, że niektóre algorytmy w odpowiedzi na rozmyte dane wejściowe generują więcej niż jedną odpowiedź. W życiu nie zawsze jest możliwe uzyskanie precyzyjnych odpowiedzi. Oczywiście precyzyjna odpowiedź zawsze jest celem, ale często trzeba się zadowolić odpowiedzią akceptowalną.

Uzyskanie jak najdokładniejszej odpowiedzi może zająć dużo czasu. Gdy otrzymasz precyzyjną odpowiedź, ale nadejdzie ona zbyt późno, aby można było z niej skorzystać, informacje staną się bezużyteczne, zatem tylko straciłeś czas. Wybór pomiędzy dwoma algorytmami, które dotyczą tego samego problemu, może sprowadzić się do wyboru pomiędzy szybkością a precyzją. Szybki algorytm może nie generować precyzyjnej odpowiedzi, ale odpowiedź nadal może być wystarczająco dobra, by dostarczać użytecznych danych wyjściowych.

Błędne odpowiedzi mogą generować problemy. Szybkie tworzenie wielu błędnych odpowiedzi jest tak samo złe jak powolne tworzenie wielu precyzyjnych odpowiedzi. W pewnym sensie celem tej książki jest pomoc w znalezieniu „złotego środka" pomiędzy odpowiedziami udzielanymi zbyt szybko i zbyt wolno oraz niedokładnymi i zbyt dokładnymi. Nawet jeśli nauczyciel matematyki naciskał na dostarczanie prawidłowych odpowiedzi w sposób, jaki był opisany w podręczniku, którego używałeś w szkole, matematyka w świecie rzeczywistym często wiąże się z dokonywaniem wyborów i podejmowaniem kompromisowych decyzji. Czasami mają one na nas zaskakujący wpływ.

Nie ma nic za darmo!

Być może słyszałeś popularny mit, że komputer może obliczyć każdy wynik, nie wkładając zbyt wielkiego wysiłku w opracowanie rozwiązania. Niestety nie istnieje absolutne rozwiązanie każdego problemu, a lepsze odpowiedzi są często dość kosztowne. Podczas pracy z algorytmami łatwo się przekonasz, że do uzyskania szybszych i bardziej precyzyjnych odpowiedzi potrzebne są dodatkowe zasoby. Rozmiar i złożoność źródeł danych, z których korzystasz, również znacząco wpływają na dokładność rozwiązania. Wraz ze wzrostem rozmiaru i złożoności problemu wzrasta ilość zasobów potrzebnych do jego rozwiązania.

Dostosowanie strategii do problemu

W części V tej książki omawiamy strategie, których można użyć, aby zmniejszyć koszty pracy z algorytmami. Najlepsi matematycy stosują sztuczki, aby uzyskać więcej wyników z wykorzystaniem mniejszej liczby komputerów. Można na przykład stworzyć złożony algorytm rozwiązania problemu lub użyć wielu prostszych algorytmów, aby rozwiązać ten sam problem za pomocą wielu procesorów. Wiele prostych algorytmów zwykle działa szybciej i lepiej niż pojedynczy złożony algorytm, nawet jeśli to podejście wydaje się sprzeczne z intuicją.

Zrozumiały opis algorytmów

Algorytmy stanowią podstawę komunikacji między ludźmi, nawet jeśli osoby te mają różne perspektywy i mówią różnymi językami. Na przykład twierdzenie Bayesa (prawdopodobieństwo wystąpienia zdarzenia przy danych przesłankach — zwięzły opis tego niesamowitego twierdzenia można znaleźć na stronie *https://betterexplained.com/articles/an-intuitive-andshort-explanation-of-bayes-theorem/*):

```
P(B|E) = P(E|B)*P(B)/P(E)
```

wygląda tak samo bez względu na to, czy mówisz po polsku, angielsku, hiszpańsku, chińsku, niemiecku, francusku czy w dowolnym innym języku. Niezależnie od tego, w jakim języku mówisz, przy takich samych danych wejściowych algorytm wygląda tak samo i działa tak samo. Algorytmy pomagają przekraczać

wszelkiego rodzaju podziały pomiędzy ludźmi, ponieważ umożliwiają wyrażanie pojęć w postaci, którą każdy może udowodnić. Przeglądając tę książkę, odkryjesz piękno i magię stosowania algorytmów do przekazywania innym nawet bardzo subtelnych myśli.

Oprócz uniwersalnych notacji matematycznych algorytmy wykorzystują języki programowania jako sposób wyjaśniania i komunikowania rozwiązywanych przez siebie wzorów. Implementacje algorytmów są dostępne w C, C ++, Javie, Fortranie, Pythonie (tak jak w tej książce), a także w innych językach. Niektórzy autorzy korzystają z pseudokodu, bo dzięki temu można zapobiec problemom związanym z prezentowaniem algorytmu w języku, który nie wszyscy znają. Pseudokod to sposób opisywania operacji na komputerze przy użyciu opisowego języka.

Stawianie czoła trudnym problemom

Ważną kwestią podczas pracy z algorytmami jest możliwość stosowania ich do rozwiązywania problemów o dowolnej złożoności. Algorytm nie myśli, nie ma emocji ani nie obchodzi go, w jaki sposób go używasz (a nawet nadużywasz). Możesz używać algorytmów w dowolny sposób wymagany do rozwiązania problemu. Na przykład grupa algorytmów wykorzystywanych do rozpoznawania twarzy (stosowana jako alternatywa dla komputerowych haseł) może się przyczynić do ujawnienia czających się na lotnisku terrorystów lub znalezienia wędrującego po ulicach zagubionego dziecka. Ten sam algorytm ma różne zastosowania. Sposób jego wykorzystania zależy od zainteresowań użytkownika. Jednym z powodów, dla których warto uważnie przeczytać tę książkę, jest pomoc w rozwiązaniu trudnych problemów, które mogą wymagać jedynie prostego algorytmu.

Strukturyzacja danych w celu uzyskania rozwiązania

Ludzie myślą o danych w niespecyficzny sposób. Stosują do tych samych danych różne reguły i rozumieją je w sposób, którego komputery nie potrafią. Komputer widzi dane w sposób ustrukturyzowany, prosty, bezkompromisowy i zdecydowanie niekreatywny. Gdy ludzie przygotowują dane do wykorzystania przez komputer, te dane często wchodzą w nieoczekiwane interakcje z algorytmami i powodują niepożądane rezultaty. Problem polega na tym, że człowiek nie dostrzega ograniczeń postrzegania danych w sposób charakterystyczny dla komputera. W poniższych punktach opisujemy dwa aspekty danych zilustrowane w wielu rozdziałach.

Zrozumienie punktu widzenia komputera

Komputer postrzega dane w prosty sposób, a jednocześnie w taki, którego ludzie zazwyczaj nie rozumieją. Zacznijmy od tego, że dla komputera wszystkie dane są liczbami. Komputery nie potrafią działać z danymi żadnego innego rodzaju. Ludzie widzą znaki na ekranie komputera i zakładają, że komputer przetwarza dane, korzystając ze znaków. Komputer nie rozumie jednak danych ani wynikających z nich implikacji. Litera *A* dla komputera to po prostu liczba 65. Właściwie nie jest to nawet liczba 65. Komputer „widzi" ciąg impulsów elektrycznych równych wartości binarnej 0100 0001.

Komputery nie rozumieją również pojęcia wielkich i małych liter. Dla człowieka mała litera *a* to po prostu inna forma wielkiej litery *A*, ale dla komputera są to dwie różne litery. Mała litera *a* to dla komputera liczba 97 (wartość binarna 0110 0001).

Jeśli te proste porównania pojedynczych liter mogą powodować takie problemy pomiędzy ludźmi a komputerami, nietrudno wyobrazić sobie, co się dzieje, gdy ludzie zaczynają tworzyć założenia dotyczące innych danych. Na przykład komputer nie potrafi słyszeć muzyki ani się nią zachwycać. Pomimo to muzyka płynie z głośników komputera. To samo dotyczy grafiki. Zamiast grafiki przedstawiającej piękną scenerię krajobrazu komputer widzi ciąg zer i jedynek.

Podczas korzystania z algorytmów ważne jest postrzeganie danych z perspektywy komputera. Komputer widzi tylko zera i jedynki, nic więcej. W konsekwencji, kiedy zaczynasz analizować algorytmy, musisz postrzegać dane w taki sam sposób. Warto wiedzieć, że postrzeganie danych w taki sposób, w jaki widzi je komputer, sprawia, że niektóre rozwiązania stają się łatwiejsze, a nie trudniejsze do znalezienia. Więcej informacji na temat tej osobliwości uzyskasz, przeglądając dane w dalszej części tej książki.

Układ danych robi różnicę

Komputery mają również ścisłe pojęcie o formacie i strukturze danych. Kiedy zaczniesz pracować z algorytmami, przekonasz się, że znacząca część zadania polega na wyświetlaniu danych w formacie, którego komputer może użyć podczas korzystania z algorytmu po to, by znaleźć rozwiązanie problemu. Chociaż człowiek potrafi widzieć wzorce w danych, które nie są dokładnie ułożone, to aby komputery znalazły ten sam wzorzec, wymagają danych w precyzyjnej postaci. Dzięki tej precyzji komputery mogą odkrywać nowe wzorce. Jest to jeden z głównych powodów stosowania algorytmów w komputerach — aby zlokalizować nowe wzorce i użyć tych wzorców do wykonywania innych zadań. Na przykład komputer może rozpoznać wzorzec wydatków klienta, a następnie wykorzystać te informacje do zwiększenia sprzedaży.


PROGRAM
PARTNERSKI
GRUPY HELION
1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ
Zmień swoją stronę WWW w działający bankomat!
Dowiedz się więcej i dołącz już dzisiaj!
http://program-partnerski.helion.pl
GRUPA
Helion

Myśl za pomocą algorytmów

Ten jasny i przystępny przewodnik pokazuje, w jaki sposób algorytmy wpływają na nasze codzienne życie — od interakcji online po osobistą komunikację. Są również niezwykle ważne, jeśli chodzi o podejmowanie różnego rodzaju decyzji. Jeśli chcesz wiedzieć, jak korzystać z procedur rozwiązywania problemów w prawdziwym świecie, książka *Algorytmy dla bystrzaków* zagwarantuje Ci doskonałe wprowadzenie do tej fascynującej, wszechobecnej dziedziny.

W książce:

- Operacje na danych
- Projektowanie algorytmów
- Podstawy teorii grafów
- Zarządzanie danymi o dużej objętości
- Upraszczanie złożonych algorytmów

John Paul Mueller — autor ponad 100 książek i 600 artykułów na różne tematy — od pracy w sieci po uczenie maszynowe.

Luca Massaron jest inżynierem danych specjalizującym się w organizowaniu i interpretowaniu danych typu big data oraz przekształcaniu ich w smart data.

dla bystrzaków

Zamówienia telefoniczne:

0 801 339900 0 601 339900

septem
septem.pl

Sprawdź najnowsze promocje:
http://dlabystrzakow.pl/promocje
Książki najchętniej czytane:
http://dlabystrzakow.pl/bestsellery
Zamów informacje o nowościach:
http://dlabystrzakow.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: rady@dlabystrzakow.pl
http://dlabystrzakow.pl

Helion

Cena 59,00 zł

ISBN 978-83-283-6076-1

9 788328 360761