Algorytmika - pojęcia podstawowe

algorytmika (ang. algorithmics)

Dział informatyki zajmujący się poszukiwaniem, konstruowaniem i badaniem własności algorytmów, w kontekście ich przydatności do rozwiązywania problemów za pomocą komputerów. Nazwa ta została wprowadzona przez izraelskiego informatyka Davida Harela w tytule jego książki o algorytmach Algorithmics. The Spirit of Computing (Algorytmika. Rzecz o istocie informatyki, Warszawa 1992, WNT). Książka ta jest zapisem pogadanek autora o algorytmice i informatyce, wygłaszanych w radiu izraelskim.

algorytm (ang. algorithm)

Precyzyjny opis sposobu rozwiązania określonego zadania lub osiągnięcia jakiegoś celu. Pierwsze algorytmy w szkole pojawiają się na lekcjach matematyki jako opisy wykonania działań na dowolnych liczbach, np. algorytm pisemnego dodawania dwóch dowolnych liczb lub algorytm pisemnego mnożenia dwóch dowolnych liczb. Wykonawcą algorytmu może być człowiek lub komputer. Algorytm jest podstawowym pojęciem informatyki. Każdy program komputerowy jest zapisem jakiegoś algorytmu. Algorytm jest tworzony na podstawie specyfikacji problemu, który ma rozwiązywać. Dane i wyniki ze specyfikacji są jednocześnie danymi wejściowymi i danymi wyjściowymi (czyli wynikami działania) algorytmu. Od algorytmu wymaga się, aby wszystkie jego elementy (dane, polecenia, wyniki) były dobrze określone, a sam algorytm był uniwersalny. Algorytm może być zapisany słownie w postaci listy kroków lub w jednym z języków programowania - wtedy jest zrozumiały dla komputera. Może być również przedstawiony w postaci graficznej, jako schemat blokowy. Oceną szybkości działania algorytmu jest jego złożoność, czyli liczba wykonywanych operacji. W praktyce często zadowalamy się oceną efektywności działania algorytmu, czyli jak szybko działa algorytm dla konkretnych danych. Mimo że komputery są coraz szybsze, istnieją problemy, których nadal nie można rozwiązać ze względu na bardzo dużą złożoność opracowanych dla nich algorytmów.

Słowo "algorytm" pochodzi od brzmienia fragmentu nazwiska arabskiego matematyka Muhammada ibn Musa al-Chorezmiego, żyjącego na przełomie VIII i IX wieku. Uważany jest on za prekursora obliczeniowych metod w matematyce. Za najwcześniejszy algorytm uznawany jest algorytm Euklidesa.

Drzewo algorytmu (ang. algorithm tree)

Graficzna reprezentacja algorytmu przyjmująca postać drzewa. W takiej reprezentacji, wierzchołki pośrednie drzewa zawierają wykonywane w algorytmie operacje, a w wierzchołkach końcowych znajdują się wszystkie możliwe wyniki wykonania algorytmu.

Drzewo binarne

Drzewo binarne (angielskie binary tree), struktura danych określona na skończonym zbiorze węzłów, którą można opisać rekurencyjnie w sposób następujący:
1) nie zawiera żadnych węzłów (drzewo puste);
2) składa się z trzech rozłącznych zbiorów węzłów: korzenia, lewego poddrzewa drzewa binarnego i prawego poddrzewa drzewa binarnego. Węzeł drzewa binarnego (każdy lub większość) zawiera wskaźnik do ojca (węzła nadrzędnego) oraz do prawego i lewego syna (węzłów podrzędnych), czyli co najwyżej dwa następniki. Pełne drzewo binarne ma w każdym wierzchołku 0 lub 2 następniki

Iteracja, pętla (ang. iteration)

Instrukcja powtarzania pewnego zbioru lub ciągu operacji. Liczba powtórzeń może być ustalona przed wykonaniem instrukcji lub może zależeć od spełnienia pewnego warunku, który jest sprawdzany w każdej iteracji.

Rekurencja

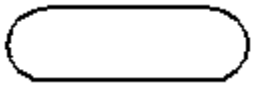
Sposób wykonywania obliczeń, w którym wydzielony podprogram wywołuje siebie samego. Rekurencję można również zastosować do opisu czynności, które nie są obliczeniami. Klasyczny jest już opis jedzenia kaszki podany przez informatyka rosyjskiego Andrieja P. Jerszowa: Jedz kaszkę znaczy weź łyżkę kaszki i jedz kaszkę (dalej). Podobnie można określić, co to znaczy tańczyć: Tańcz to zrób jeden krok i tańcz (dalej).
Rekurencyjny opis obliczeń jest na ogół bardziej zwarty niż opis tych samych obliczeń bez użycia rekurencji. Taki opis jest stosowany np. przy opisie fraktali, które są nawet definiowane jako twory podobne do swoich części.

Schemat blokowy

Schematy blokowe są tzw. metajęzykiem. Oznacza to, że jest to język bardzo ogólny, służy do opisywania algorytmów w taki sposób, by na jego podstawie można było je zaimplementować w każdym języku.

Częściami składowymi schematów blokowych są proste figury geometryczne, np. prostokąt, romb, koło, równoległobok itd... W tych figurach umieszczamy warunki oraz proste instrukcje, przy czym mogą być one związane z jakimś konkretnym językiem (np. symbolem instrukcji przypisania może być ":=" tak, jak w Pascalu lub "=" tak, jak w C) Jeśli tworząc schemat nie jesteśmy jeszcze zdecydowani w jakim języku będziemy pisali nasz program lub tworzymy schemat dla kogoś, to lepiej jest stosować notację bardziej symboliczną, np. instrukcję przypisania zapisywać jako strzałkę skierowaną od wartości przypisywanej do zmiennej. Za chwilę przedstawię elementy składowe schematów blokowych.

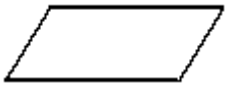
⟶ Poszczególne elementy schematu łączy się za pomocą strzałek. W większości przypadków blok ma jedną strzałkę wchodzącą i jedną wychodzącą, lecz są także wyjątki



Ta figura oznacza początek lub koniec algorytmu. W każdym algorytmie musi się znaleźć dokładnie jedna taka figura z napisem "Start" oznaczająca początek algorytmu oraz dokładnie jedna figura z napisem "Stop" oznaczająca koniec algorytmu. Najczęściej popełnianym błędem w schematach blokowych jest umieszczanie kilku stanów końcowych, zależnych od sposobu zakończenia programu. Jest to niedopuszczalne, w programie mamy przecież dokładnie jedną instrukcję "end." Blok symbolizujący początek algorytmu ma dokładnie jedną strzałkę wychodzącą a blok symbolizujący koniec ma co najmniej jedną strzałkę wchodzącą.

Jest to figura oznaczająca proces. W jej obrębie umieszczamy wszelkie obliczenia lub podstawienia. Proces ma dokładnie jedną strzałkę wchodzącą i dokładnie jedną strzałkę wychodzącą.

Romb symbolizuje blok decyzyjny. Umieszcza się w nim jakiś warunek (np. "x>2"). Z dwóch wybranych wierzchołków rombu wyprowadzamy dwie możliwe drogi: gdy warunek jest spełniony (strzałkę wychodzącą z tego wierzchołka należy opatrzyć etykietą "Tak") oraz gdy warunek nie jest spełniony. Każdy romb ma dokładnie jedną strzałkę wchodzącą oraz dokładnie dwie strzałki wychodzące.



Równoległobok jest stosowany do odczytu lub zapisu danych. W jego obrębie należy umieścić stosowną instrukcję np. Read(x) lub Write(x) (można też stosować opis słowny np. "Drukuj x na ekran"). Figura ta ma dokładnie jedną strzałkę wchodzącą i jedną wychodzącą.

Ta figura symbolizuje proces, który został już kiedyś zdefiniowany. Można ją porównać do procedury, którą definiuje się raz w programie, by następnie móc ją wielokrotnie wywoływać. Warunkiem użycia jest więc wcześniejsze zdefiniowanie procesu. Podobnie jak w przypadku zwykłego procesu i tu mamy jedno wejście i jedno wyjście.

Koło symbolizuje tzw. łącznik stronicowy. Może się zdarzyć, że chcemy "przeskoczyć" z jednego miejsca na kartce na inne (np. by nie krzyżować strzałek). Możemy w takim wypadku posłużyć się łącznikiem. Umieszczamy w jednym miejscu łącznik z określonym symbolem w środku (np. cyfrą, literą) i doprowadzamy do nie go strzałkę. Następnie w innym miejscu kartki umieszczamy drugi łącznik z takim samym symbolem w środku i wyprowadzamy z niego strzałkę. Łącznik jest często porównywany do teleportacji (z jednego miejsca na kartce do drugiego). Łączniki występują więc w parach, jeden ma tylko wejście a drugi wyjście.

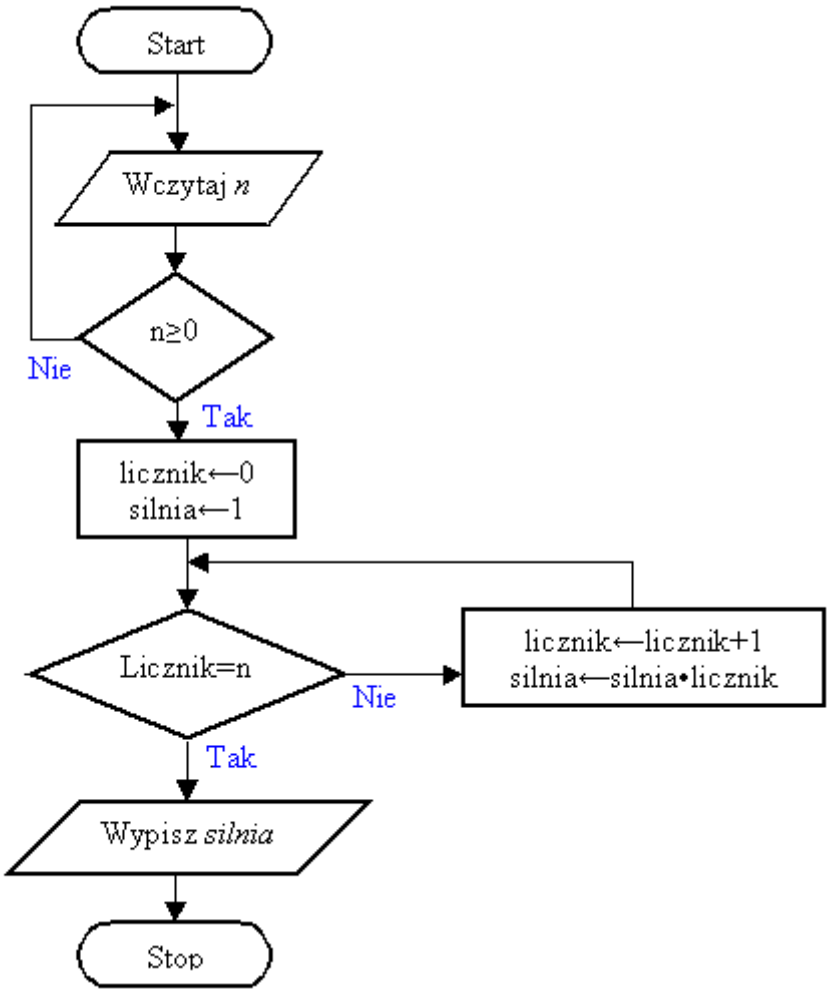
Ten symbol to łącznik międzystronicowy. Działa analogicznie jak pierwszy, lecz nie w obrębie strony. Przydatne w złożonych algorytmach, które nie mieszczą się na jednej kartce. Uwaga: jeśli stosujemy oba typy łączników w schemacie, to najlepiej jest stosować liczby do

identyfikowania jednych i litery do drugich. Dzięki temu nie dojdzie do pomyłki.

Teraz opiszemy przykładowy algorytm za pomocą schematu blokowego. Zdefiniujmy iteracyjną wersję silni. Dla przypomnienia: rekurencyjna definicja silni wygląda następująco

silnia(0)=1

silnia(n)=n*silnia(n-1)



Program, program komputerowy (ang. computer program)

Zestaw poleceń przeznaczonych do wykonania przez komputer, powodujących jego działanie zgodnie z zamierzeniem użytkownika. Program może być zapisany w jednym z języków

programowania, ale przed wykonaniem go przez komputer musi zostać przetłumaczony na kod maszynowy, czyli ciąg rozkazów dla procesora. Można wyróżnić programy: systemowe (niezbędne do poprawnego działania systemu komputerowego), narzędziowe (np. programy antywirusowe), użytkowe (służące użytkownikowi do wykonywania konkretnych zadań, np. rysowania, pisania), edukacyjne. Jest jeszcze wiele innych kategorii oprogramowania, np. gry komputerowe, języki programowania itd.

Wyszukiwanie binarne

wyszukiwanie binarne, przeszukiwanie binarne (ang. binary search), wyszukiwanie metodą połowienia, przeszukiwanie metodą połowienia, metoda połowienia (ang. binary search). Metoda (algorytm) przeszukiwania zbioru uporządkowanego w celu znalezienia wybranego elementu, np. hasła w słowniku. W każdym jej kroku, pozostały do przeszukania zbiór jest dzielony na dwie części, (na ogół na połowy), z których jedna zawiera poszukiwany element, a druga go nie zawiera. W tym celu środkowy element zbioru jest porównywany z poszukiwanym elementem i do dalszych poszukiwań jest pozostawiana ta część, w której może się on znajdować. Dzięki temu w jednym kroku tej metody przeszukiwany zbiór ulega zmniejszeniu o połowę i bardzo szybko trafiamy na poszukiwany element lub stwierdzamy, że nie ma go w tym zbiorze. Na przykład w taki sposób poszukujemy na ogół hasła w słowniku, numeru telefonu w książce telefonicznej, czy danych o książce w katalogu. Jeśli słownik ma 1000 stron, to stosując tę metodę każde zawarte w nim hasło można znaleźć po przejrzeniu co najwyżej 10 stronic. Ta metoda jest szczególnym przypadkiem metody dziel-i-zwyciężaj.

Złożoność algorytmu (ang. algorithm complexity)

Liczba elementarnych operacji, np. porównań, dodawań, mnożeń, przestawień elementów, wykonywanych przez algorytm, podawana na ogół w zależności od długości (ilości) danych. Na przykład liczba porównań potrzebnych do znalezienia najmniejszego (lub największego) elementu w nieuporządkowanym zbiorze jest o jeden mniejsza od liczby elementów w zbiorze. A w algorytmie porządkowania przez wybór jest wykonywanych n(n-1)/2 porównań i n-1 przestawień elementów, gdy porządkowany jest zbiór złożony z n elementów. Rzeczywisty czas działania algorytmu może być podany dopiero dla jego konkretnej realizacji, przeznaczonej dla konkretnego komputera. Inną miarą jakości algorytmu jest jego efektywność określana na podstawie testowania szybkości jego działania na przykładowych

danych. Złożoność i efektywność można również zdefiniować do programów.

Algorytm Newtona-Raphsona (ang. Newton-Raphson algorithm)

Algorytm służący do obliczania wartości pierwiastka kwadratowego metodą iteracyjną. Dla danej liczby podpierwiastkowej a, obliczenia wartości sqrt(a) rozpoczynają się od wartości początkowej x0, która jest przybliżeniem rozwiązania, za którą można przyjąć na przykład (1+a)/2. Następne przybliżenia są wyznaczane ze wzoru: xi+1 = (xi+a/xi)/2. Obliczenia kończą się, gdy dwa kolejne przybliżenia niewiele różnią się między sobą, czyli gdy różnica |xi - xi+1| jest małą liczbą, na przykład 0.000001.

Algorytm aproksymacyjny (ang. approximation algorithm), algorytm przybliżony

Algorytm umożliwiający otrzymanie przybliżonego rozwiązania, czyli takiego rozwiązania, które nie jest najlepsze (tj. optymalne), ale różni się od niego niewiele w sensie przyjętej miary dokładności. Algorytmy przybliżone są stosowane wtedy, gdy nie potrafimy znaleźć rozwiązania optymalnego i musimy się zadowolić rozwiązaniem przybliżonym. Ponadto algorytmy komputerowe działające na liczbach reprezentowanych w komputerze w sposób przybliżony dają również rozwiązania tylko przybliżone z pewną dokładnością. Algorytm Newtona-Raphsona, służący do obliczania wartości pierwiastka kwadratowego, jest przykładem algorytmu przybliżonego.

Aproksymacja

Aproksymacja, przybliżenie wielkości dokładnej przez inną, bliską jej w pewnym określonym sensie. Mówi się o aproksymacji krzywej przez łamaną, liczby niewymiernej przez wymierną, dowolnej funkcji ciągłej przez wielomian.

Algorytm z nawrotami (ang. backtracking algorithm), przeszukiwanie z nawrotami

Algorytm polegający na poszukiwaniu rozwiązania wśród wszystkich możliwych rozwiązań w sposób, który gwarantuje, że nie zostanie ono przeoczone, jeśli tylko istnieje. Przykładem takiego algorytmu może być sposób poszukiwania wyjścia z labiryntu. Przyjmuje się w tym przypadku, że przejście z danego punktu do następnego punktu wykonujemy w ustalonym porządku możliwych punktów do przejścia (np. od lewej do prawej) i gdy nie ma już żadnej możliwości pójścia dalej, wówczas cofamy się (tj. robimy nawrót) do punktu, z którego przyszliśmy. Tak poruszał się po labiryncie mityczny Tezeusz w poszukiwaniu potwora

Minotaura, a w nawrotach pomagała mu nić ofiarowana mu przed wejściem do labiryntu przez Ariadnę.

Algorytm zachłanny (ang. greedy algorithm)

Algorytm, w którym na każdym kroku podejmowana jest możliwie najlepsza decyzja. Jeśli chcemy osiągnąć jak najwięcej, to w myśl strategii zachłannej na każdym kroku bierzemy najwięcej, jak to tylko możliwe. Z kolei, by wziąć możliwie najmniej czegoś, na każdym kroku powinniśmy brać jak najmniej. Przykładem strategii zachłannej jest metoda wydawania reszty w postaci najmniejszej liczby banknotów i monet. W myśl tej strategii sprzedawca powinien wydawać resztę od największych banknotów, próbując wydać możliwie najwięcej banknotów jeszcze mieszczących się w reszcie. Na przykład resztę w wysokości 9 złotych i 93 groszy powinniśmy otrzymać w postaci sumy monet: 5+2+2 złotych oraz 50+20+20+2+1 groszy.

Ta strategia jest zachłanna dla obu stron: klient otrzymuje resztę w postaci najmniejszej liczby banknotów, a sprzedawca ma mniej kłopotu z wydawaniem (dzięki czemu również ma mniej okazji, by pomylić się w wydawaniu).

Etapy rozwiązywania problemów:

Sformułowanie zadania.
Określenie danych wejściowych.
Określenie celu, czyli wyniku.
Poszukiwanie metody rozwiązania, czyli algorytmu w postaci:
Przedstawienie algorytmu w postaci:
- opisu słownego
- listy kroków
- schemat blokowego
- jednego z języków programowania

Analiza poprawności rozwiązania.
Testowanie rozwiązania dla różnych danych - ocena efektywności przyjętej metody.