

Wybrane metody projektowania algorytmów

Algorytm to:

- ◆ skończony ciąg/sekwencja reguł, które aplikuje się na skończonej liczbie danych, pozwalający rozwiązywać zbliżone do siebie klasy problemów
- ◆ zespół reguł charakterystycznych dla pewnych obliczeń lub czynności informatycznych

Samo pochodzenie terminu algorytm nie zawsze było do końca jasne. Dopiero specjaliści zajmujący się historią matematyki odnaleźli najbardziej prawdopodobny źródłosłów: termin ten pochodzi od nazwiska perskiego pisarza matematyka Muhammada ibn Musy al-Chuwarizmiego¹ (żył w IX wieku n.e.). Jego zasługą jest dostarczenie klarownych reguł wyjaśniających krok po kroku zasady operacji arytmetycznych wykonywanych na liczbach dziesiętnych.

Słowo algorytm jest często łączone z imieniem greckiego matematyka Euklidesa (365 - 300 p.n.e.) i jego słynnym przepisem na obliczanie największego wspólnego dzielnika dwóch liczb a i b (NWD):

dane wejściowe: a i b , zmienne pomocnicze: c , res

dopóki $a > 0$ wykonuj:

podstaw za c resztę z dzielenia a przez b :

podstaw za b liczbę a :

podstaw za a liczbę c :

podstaw za res liczbę b :

rezultat: res.

¹ Jego nazwisko pisane było po łacinie jako *Algorismos*.

Podany powyżej przepis na obliczanie NWD można przykładowo zapisać jako:

```
NWD (dane wejściowe: a i b) {  
  jeśli b równa się zero, to  
    rezultat: a  
  w przeciwnym wypadku  
    rezultat: NWD(b, reszta z dzielenia a przez b)  
}
```

To już zaczyna powoli przypominać język programowania! Algorytm posiada parametry i wywołuje sam siebie...

Czym powinien się charakteryzować algorytm?

Popatrzmy na przykład swego rodzaju algorytmu, pochodzącego z doskonale znanego gatunku „książka kucharska”. Każdy z nas z pewnością umie zaparzyć kawę według podanego przepisu:

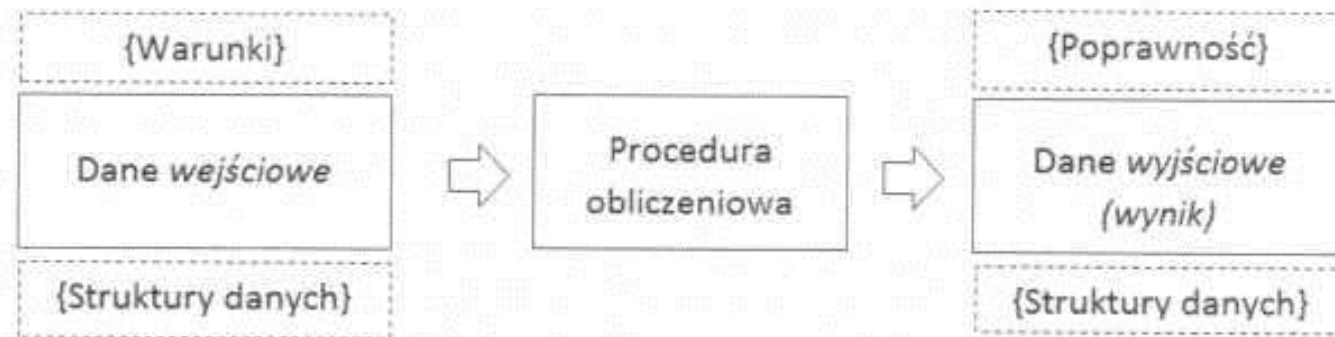
- ◆ włączyć gaz;
- ◆ zagotować niezbędną ilość wody;
- ◆ wsypać zmieloną kawę do szklanki;
- ◆ zalać kawę wrzącą wodą;
- ◆ osłodzić do smaku;
- ◆ poczekać, aż odpowiednio naciągnie.

Używając nieco bardziej naukowego żargonu, można powiedzieć, że przepis przygotowania kawy w powyższej formie nie jest deterministyczny, połączenie pewnej swobody sformułowań danych wejściowych (kawa, woda, cukier, temperatura...) z ich możliwą realizacją (niejasny czas przygotowania) powoduje, że wynik końcowy może być za każdym razem różny, pomimo pozornie tych samych danych wejściowych.

Każdy algorytm:

- ◆ posiada dane wejściowe (w ilości większej lub równej zero) pochodzące z dobrze zdefiniowanego zbioru, omawiany algorytm Euklidesa operuje na dwóch liczbach całkowitych, ale napotkasz także algorytmy operujące na tekstach (napisy lub ciągi znaków) lub bardzo złożonych strukturach wejściowych (grafy, zbiory);
- ◆ produkuje pewien wynik (niekoniecznie numeryczny);
- ◆ jest precyzyjnie zdefiniowany (każdy krok algorytmu musi być jednoznacznie określony);
- ◆ daje się zastosować do rozwiązywania całej klasy zagadnień, a nie tylko jednego konkretnego zadania (np. algorytm sortowania powinien równie dobrze działać na tablicy liczb całkowitych, jak i na tablicy obiektów złożonych).

Rysunek 1.1.
Algorytm informatyczny



Na rysunku 1.1 symbolicznie przedstawiono główne cechy algorytmu rozumianego w kontekście informatycznym: pewne dane wejściowe są przekształcane przez procedury obliczeniowe w celu uzyskania danych wyjściowych (lub po prostu wyników).

Realizację algorytmu wspomaga odpowiednie modelowanie z użyciem:

- ◆ warunków wejściowych mających na celu m.in. wyeliminowanie danych, które nie zawierają się w domenie obsługiwanej przez algorytm (np. pewien algorytm może akceptować wyłącznie dodatnie liczby całkowite);
- ◆ struktur danych pozwalających na przechowywanie i obsługę danych przetwarzanych przez algorytm (np. tablice, listy, drzewa).

Co oznacza poprawność algorytmu? Można tu uwypuklić następujące cechy.

- ◆ Algorytm powinien być skończony (wynik algorytmu musi zostać kiedyś dostarczony). Inaczej mówiąc, dla algorytmu A i danych wejściowych D powinno być możliwe precyzyjne określenie czasu wykonania $T(A, D)$.
- ◆ Algorytm zazwyczaj jest deterministyczny, gdyż w 99% przypadków zależy nam na tym, aby wynik jego działania był jednoznacznie określony przez warunki początkowe (parametry). Oczywiście algorytm niedeterministyczny może być poprawny, jeśli wynika to z samych cech algorytmu. Co należy zrobić, aby algorytm komputerowy stał się niedeterministyczny? Najprościej wprowadzić do niego czynnik losowości (np. generator liczb losowych) albo zastosować przetwarzanie równoległe. Ekstremalnym przykładem algorytmów niedeterministycznych są te, które działają na komputerach kwantowych, nieużywających logiki zero-jedynkowej, tylko stanów pośrednich! Warto mieć świadomość, że algorytmika czasami wchodzi w obszary znacznie odbiegające od stereotypowych zastosowań „komputerowych”.
- ◆ Niecierpliwość każe nam szukać algorytmów efektywnych, tzn. wykonujących swoje zadanie w jak najkrótszym czasie i wykorzystujących do tego celu jak najmniejszą ilość. Ten ostatni warunek jest istotny, gdyż w praktyce algorytmy są realizowane przez złożone procedury działające na rzeczywistych komputerach, posiadających ograniczoną pamięć i moc obliczeniową.