

## Notatka do modułu 2 - Algorytmy

- **Algorytm** - jednoznacznie sformułowany sposób postępowania, który w skończonej liczbie kroków umożliwia rozwiązywanie określonego zadania. Zakodowany w wybranym języku programowania zamienia się w program komputerowy. Inaczej jest to sformalizowany zestaw czynności, które należy wykonać aby uzyskać dany wynik, w szczególności, aby wyliczyć wyniku na podstawie określonych danych.

- **Sposoby przedstawiania algorytmów:**

- a) **Opis słowny** - określa on zazwyczaj ogólny schemat działania bez wyliczania kroków. Zapis dokonywany jest w postaci słów

- b) **Lista kroków** - przedstawia algorytm w postaci punktów, które wykonywane krok po kroku (a czasami z użyciem pętli) prowadzą do rozwiązania zadania

- c) **Schemat blokowy** - graficzny sposób zapisu algorytmu, gdzie z użyciem figur geometrycznych definiuje się określone typy instrukcji oraz połączenia.

- d) **Drzewo algorytmiczne (decyzyjne)** - graficzna metoda wspomaganie procesu decyzyjnego. Algorytm drzew decyzyjnych jest również stosowany w uczeniu maszynowym do pozyskiwania wiedzy na podstawie przykładów.

- e) **Język algorytmiczny (programowania)** - składa się on ze zbiorów reguł syntaktycznych oraz semantyki, które opisują, jak należy budować poprawne wyrażenia oraz jak komputer ma je rozumieć. Jest to usystematyzowany sposób na precyzyjny zapis algorytmów oraz innych zadań.

- **Cechy algorytmu:**

- **Jednoznaczność** (określoność) - wykonanie wszystkich procesów prowadzi do uzyskania poprawnego wyniku końcowego; dobry algorytm zawiera zabezpieczenia, nie dopuszczające do jego przerwania w trakcie realizacji z powodu zaistnienia nieprzewidzianych okoliczności; wiąże się ze skończonością liczby operacji

- **Uniwersalność** (masowość) - algorytm ma rozwiązywać wszystkie problemy danego typu

- **Szczegółowość** - polega na tym aby tak rozpisać czynności (na czynności elementarne), aby mogły być odczytane i wykonane przez realizatora algorytmu (czyli użytkownika)

- **Sekwencyjność** - dotyczy kolejności wykonania operacji

- **Złożoność** (czasowa i pamięciowa) - algorytm "zabiera" zasoby niezbędne do wykonania czynności (pamięć komputera, czas, ilość włożonej energii etc.)

- **"Słownik" algorytmu:**

Niezwykle ważnym aspektem, który należy brać pod uwagę podczas tworzenia i realizowania algorytmu jest znajomość odpowiedniego "języka". Musi być on znany i zrozumiały zarówno dla twórcy, jak i odbiorcy algorytmu. Jak podaje Wikipedia (o tym dlaczego twórcy tej strony lubią Wikipedię, opowiemy na zajęciach ;-)) typowym algorytmem przedstawianym jako przykład stosowanego w życiu codziennym jest przepis kulinarny. Cyt. "Dla przykładu, aby ugotować bigos należy w określonej kolejności oraz odstępach czasowych (...) dodawać właściwe rodzaje kapusty i innych składników. (...) Przykład ten ma wyłącznie charakter poglądowy, ponieważ język przepisów kulinarnych nie został jasno zdefiniowany. Algorytmy zwykle formułowane są w sposób ścisły w oparciu o język matematyki."

Dodać należy, że aby poprawnie wykonać ten algorytm (ugotować bigos) konieczna jest właśnie znajomość wyrażen z tzw. "słownika". Gotujący tą potrawę musi wiedzieć czym jest obieranie, krojenie i gotowanie na wolnym ogniu. Musi także znać podstawowe jednostki czasu: minuty,

## Notatka do modułu 2 - Algorytmy

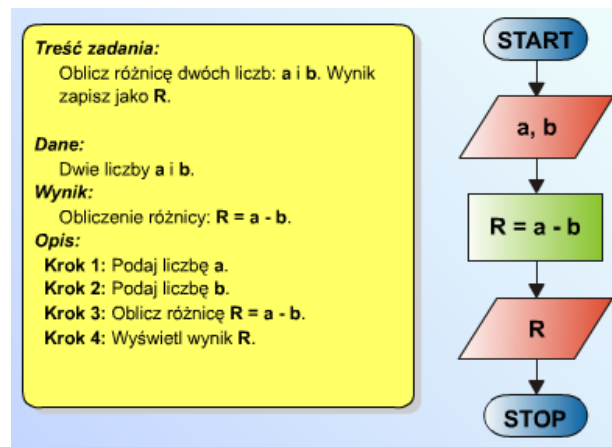
godziny etc. Niestety, w kuchni nawet perfekcyjne wykonanie algorytmu nie zawsze skutkuje ugotowaniem pysznego obiadu ;-)

### • Podstawowy podział algorytmów:

- **algorytm liniowy** - składa się z operacji wykonywanych kolejno jedna po drugiej (sekwencyjnie). Tak ustalony porządek wykonywania działań określa tylko jeden sposób postępowania

- **algorytm złożony (z rozgałęzieniami)** - sposób postępowania, w którym wyróżniamy miejsce, gdzie wybór dalszej drogi działania uzależniony jest od spełnienia lub nie pewnego warunku albo wielu warunków.

### Przykład algorytmu liniowego



### Przykład algorytmu złożonego

