

Algorytmy – wprowadzenie

Algorytm - co to jest?

Słowo Algorytm oznacza „Zbiór skończonych reguł lub instrukcji, których należy przestrzegać w obliczeniach lub innych operacjach rozwiązywania problemów” lub „Procedurę rozwiązywania problemu matematycznego w skończonej liczbie kroków, która często obejmuje operacje rekurencyjne”.

Prostszymi słowami:

Algorytm jest przepisem opisującym krok po kroku rozwiązanie problemu lub osiągnięcie jakiegoś celu.

Filmik pomocniczy:

- KhanAcademy – <https://youtu.be/WsCeeFjWiHE>

Zabawa w zgadywanie liczby

Zagrajmy w prostą grę, która pozwoli Ci zrozumieć, dlaczego różne algorytmy, zastosowane do rozwiązania tego samego problemu, mogą mieć tak bardzo różną skuteczność. Komputer wylosuje liczbę całkowitą od 1 do 15. Twoje zadanie polega na tym, żeby zgadnąć tę liczbę. Próbujesz tak długo, aż zgadniesz, a komputer za każdym razem podpowiada Ci, czy Twoja liczba jest większa, czy mniejsza od tej, której szukasz

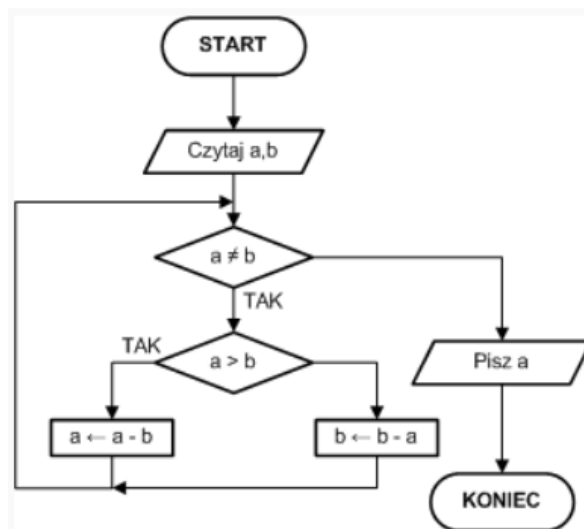
Sposoby zapisu

Opis słowny





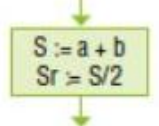
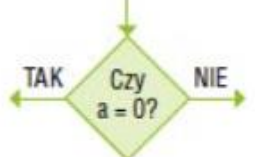


Algorytm parzenia kawy

Wziąć czajnik, sprawdzić czy jest w nim wystarczająca ilość wody, jeśli nie należy dolać wody. Postawić czajnik na ogień. Wziąć filiżankę i sprawdzić czy jest czysta. Jeśli nie, należy ją umyć. W razie potrzeby powtórzyć operacje. Wsypać do filiżanki właściwą ilość kawy i cukru. Czekać na zagotowanie wody. Gdy woda się zagotuje zalewamy przygotowaną w filiżance kawę z cukrem

Schemat blokowy



Lista elementów:

Reprezentacja graficzna	Opis operacji	Uwagi
	Początek algorytmu	Blok z napisem „Start” zaczyna algorytm. Wychodzi z niego tylko jedno połączenie i żadne do niego nie wchodzi. W jednym schemacie może funkcjonować tylko jeden taki blok.
	Zakończenie algorytmu	Blok z napisem „Koniec” kończy algorytm. Wchodzi do niego jedno połączenie, żadne nie wychodzi. W jednym schemacie może być wiele takich bloków.
	Wprowadzanie danych (blok wejścia)	Blok z napisem „Wprowadź” służy do wprowadzania danych. Ma jedno połączenie wchodzące i jedno wychodzące. W jednym schemacie może być wiele takich bloków.
	Wyprowadzanie wyników (blok wyjścia)	Blok z napisem „Wyprowadź” służy do wyprowadzania wyników. Ma jedno połączenie wchodzące i jedno wychodzące. W jednym schemacie może być wiele takich bloków.
	Wykonywanie działań (blok operacyjny)	Blok, w którym wykonywane są różne operacje, m.in. obliczenia. Ma jedno połączenie wchodzące i jedno wychodzące. W jednym bloku można wpisać więcej niż jedno wyrażenie. W jednym schemacie może być wiele takich bloków.
	Sprawdzanie warunku (blok warunkowy albo decyzyjny)	Blok podejmowania decyzji. Wchodzi do niego jedno połączenie, wychodzą dwa: • z napisem „Tak”, gdy warunek jest spełniony; • z napisem „Nie”, gdy warunek nie jest spełniony. W jednym schemacie może być wiele takich bloków.
	Łącznik	Łącznik stosuje się, gdy schemat blokowy rysujemy w kilku częściach, np. na dwóch stronach. Umieszczony wewnątrz numer powinien być taki sam w obu łączonych częściach.
	Połączenie	Połączenie łączy bloki. Tworzy je linia prosta bądź łamana, zakończona strzałką. Połączenie może dochodzić również do innego połączenia.

Lista kroków

Algorytm gotowania jajka na miękko

- Krok 1. Włóż jajko do gotującej się wody.
- Krok 2. Zanotuj czas początkowy t_0 .
- Krok 3. Odczytaj czas aktualny t .
- Krok 4. Oblicz $Dt = t - t_0$.
- Krok 5. Jeśli $Dt < 3 \text{ min.}$, to przejdź do kroku 3.
- Krok 6. Wyjmij jajko z gotującej się wody. Zakończ algorytm.

#matura

Lista kroków jest w pełni honorowanym, na maturze z informatyki w części pisemnej, sposobem przedstawiania algorytmów. Jest to alternatywa dla osób, którym język programowania sprawia problemy. Niestety na części praktycznej nie ma możliwości korzystania z tej metody.

Lista kroków polega na wypisaniu kolejnych kroków algorytmu w postaci listy np. numerowanej. Kolejne zagnieżdżone bloki należące do instrukcji **if** czy **for** opisujemy wprowadzając następne poziomy numeracji np. pierwszy poziom będzie miał numery 1, 2, 3, ..., ale blok należący do kroku trzeciego będzie numerowany następująco: 3.1, 3.2, 3.3, ..., a blok należący do kroku 3.2 numerujemy 3.2.1, 3.2.2, 3.2.3, ... itd. Możesz zastosować konwencję używaną w języku Python polegającą na wydzielaniu kolejnych bloków wprowadzając odpowiednie wcięcia np.:

```
Inicjuj zmienne a i b.  
Wczytaj wartość zmiennej a  
Wczytaj wartość zmiennej b  
Jeśli a jest większe od b to  
    wypisz tekst: "Pierwsza liczba jest większa od drugiej"  
w przeciwnym wypadku  
    Jeśli a równe b to  
        wypisz tekst: "Podane liczby są równe"  
    w przeciwnym wypadku  
        wypisz tekst: "Druga liczba jest większa od pierwszej"
```

Na co zwrócić uwagę

Tworząc iteracje (pętle), nie możesz zapisać jej zbyt ogólnikowo np.

1. Wykonaj 10 przejść pętli.

Prawidłowy zapis to:

1. Ustaw zmienną **i** na 0.
2. Dopóki **i** < 10 wykonuj kroki 2.1 oraz 2.2.
 - 2.1. [jakieś instrukcje].
 - 2.2. Zwiększ zmienną **i** o jeden.

To samo dotyczy poruszania się po indeksach tablicy:

Zbyt ogólnikowy zapis:

1. Przejdź przez wszystkie elementy tablicy i je zsumuj.

Prawidłowy zapis to:

1. Ustaw zmienną **i** na 1.
2. Zapisz wartość komórki tablicy o indeksie 0 do zmiennej **Suma**.
3. Wykonuj kroki 3.1 oraz 3.2, dopóki wartość zmiennej **i** jest mniejsze od 10.
 - 3.1. Zwiększ zmienną **Suma** o wartość komórki o indeksie **i**.
 - 3.2. Zwiększ wartość zmiennej **i** o jeden.
4. ...

Przykład

Wyznacz większą liczbę z podanych na wejściu dwóch liczb całkowitych.

0. Inicjuj zmienne a i b
1. Wczytaj wartość do zmiennej a .
2. Wczytaj wartość do zmiennej b .
3. Jeśli a jest większe od b , wykonaj krok 3.1, w przeciwnym razie wykonaj krok 3.2.
 - 3.1. Wypisz wartość zmiennej a .
 - 3.2. Wypisz wartość zmiennej b .

Drzewa decyzyjne

Jest to prosta, lecz użyteczna w uczeniu maszynowym, koncepcja opierająca się o drzewo. Bardzo często stosowana do klasyfikacji, czyli przypisanie obserwacji zbioru danych do jednej z klas.

Na każdym węźle drzewa, dokonujemy podziału zbioru na 2 lub więcej mniejsze zbiory, mając na celu, jak najlepsze odseparowanie obserwacji należących do różnych klas.

Pseudokod

Jest to nieformalny sposób zapisania algorytmu podobny do języka programowania. Słowa kluczowe oraz instrukcje są spolszczone. Bloki należące do instrukcji warunkowej, iteracyjnej czy funkcji oznaczamy odpowiednim wcięciem. Niektóre operatory, takie jak przypisanie, modulo czy dzielenie całkowite zastąpione zostały następującym zapisem:

Instrukcję warunkową możemy zapisać w następujący sposób:

```
Jeśli  $a > b$  to
    zamień( $a$ ,  $b$ )
     $b \leftarrow a \bmod b$ 
w przeciwnym razie
     $b \leftarrow b \div a$ 
```

Instrukcję iteracyjną może wyglądać w taki sposób:

```
Dla  $i \leftarrow 1$  do  $n$  powtarzaj
    wypisz( $tab[i]$ )
```

lub

```
 $i \leftarrow 1$ 
dopóki  $i < 10$  wykonuj
    wypisz( $tab[i]$ )
     $i \leftarrow i + 1$ 
```

Operator	Pseudokod	C++	Przykład
przypisania	\leftarrow	=	a \leftarrow 10
reszta z dzielenia	mod	%	a \leftarrow b mod c
dzielenie całkowite	div	/	a \leftarrow b div c
operator porównania	=	==	jeżeli a = b to
a różne b	<>	!=	jeżeli a <> b

Przykład

Przykład 1

Wczytaj dwie liczby całkowite i wyznacz większą z nich.

```
wczytaj(a)
wczytaj(b)
Jeśli a > b to
    wypisz(a)
w przeciwnym razie
    wypisz(b)
```

Zadania

1. Narysuj schemat blokowy algorytmu obliczania pola prostokąta.
2. Napisz w postaci listy kroków algorytm wysyłania wiadomości e-mail.
3. Narysuj schemat blokowy algorytmu wyświetlającego wszystkie liczby całkowite zawarte pomiędzy liczbami a i b podanymi przez użytkownika.

Rozpatrz dwa przypadki:

- a) kiedy a jest na pewno mniejsze od b
- b) kiedy a oraz b są zupełnie dowolne

4. Narysuj schemat blokowy algorytmu, który obliczy sumę liczb zawartych pomiędzy dowolnymi liczbami a oraz b podanymi przez użytkownika.