Zjazd 8

INFORMATYKA SEMESTR II

Temat: Algorytmiczne rozwiązywanie prostych problemów i ich komputerowa realizacja. Portale edukacyjne, przykłady lekcji, programów dydaktycznych.

# Jak się uczyć algorytmów i myślenia algorytmicznego?



***Algorytm to sposób rozwiązania problemu w skończonej liczbie kroków***

Czy wiesz co to wartość bezwzględna, jak zsumować kilka liczb, pamiętasz jak wykonywać mnożenie i dzielenie pisemne, obliczyć pole koła? – zapewne ta wiedza z czasów podstawówki nadal pozostaje w Twojej głowie i bez trudu przekażesz ją kilkuletniemu dziecku. Teraz musisz nauczyć się myśleć, w jaki sposób swoją wiedzę o sposobie rozwiązania problemu przekazać maszynie – twojemu komputerowi za pomocą algorytmu.

### Algorytmy i programowanie – jak zacząć?

***Jak się uczyć algorytmów? Czy można nauczyć się logicznego myślenia, myślenia algorytmicznego?***

Pytanie zadane w tytule posta pojawia się co jakiś czas w wiadomościach od opublikowania posta [**Po co programiście algorytmika?**](https://www.flynerd.pl/2018/09/po-co-programiscie-algorytmika.html). Uświadomiłam sobie, że póki co nie będę wstanie napisać wszystkich postów, które gdzieś tam mam w głowie, a jednocześnie chciałabym, by osoby, które tu wchodzą miały materiały do nauki.

Uważam, że programowanie i algorytmy to naczynia połączone. **Rozwiązywanie zadań algorytmicznych jest ważnym etapem nauki programowania** (niezależnie czy wybierzecie front-end, back-end, aplikacje mobilne czy inne dziedziny związane z programowaniem), chociaż zgadzam się, że **nie jest niezbędne by dostać pierwszą pracę w IT**, czy zostać programistą / programistką.

Tak czy inaczej zdania nie zmienię – pozostanę przy tym, że ruszanie komórek mózgowych i rozwiązywanie problemów algorytmicznych jest umiejętnością w pracy programisty ważną.

## Jak się uczyć algorytmiki?

Nauka algorytmów może wydawać się czarną magią, mimo że materiałów do nauki algorytmów jest w sieci całkiem sporo. Problem polega na tym, że jeśli nie miało się algorytmów jako przedmiotu na studiach, to nie do końca wiadomo za co się wziąć tyle tej wiedzy jest.

Tutaj, więc posłużę się własnym, prostym sylabusem:

**1) Czym są dane i jak je przechowujemy?
2) Struktury: kopiec (stóg), kolejka (FIFO), stos (LIFO), grafy i ich reprezentacje, drzewa
3) Cykl i ścieżka Eulera, cykl Hamiltona
4) Podstawy złożoności obliczeniowej (w tym notacja-O)
5) Algorytmy sortowania:**

* **insert sort – przez wstawianie,**
* **selection sort – przez wymianę**
* **quick sort – szybkie,**
* **bubble sort – bąbelkowe,**
* **heap sort – stogowe**
* **counting sort – przez zlicznanie,**
* **merge sort – przez scalanie**

**6) Programowanie dynamiczne (problem plecakowy)
7) Rekurencja
8) Tablica haszująca**

To jest przykład i to bardzo okrojony, ale w moim odczuciu wystarczający by załapać podstawy oraz je przećwiczyć. Oczywiście można wybrać też opcję Learn hard way – i skorzystać z sylabusa Algorytmy i struktury danych [„ważniaka”](http://wazniak.mimuw.edu.pl/index.php?title=Algorytmy_i_struktury_danych) (jest to materiał akademicki, bardzo rozbudowany, ale świetnie opracowany i wśród studentów informatyki wręcz kultowy).

## Materiały do nauki algorytmiki?

Oczywiście wszystkie te pojęcia można sobie wygooglować i je po prostu przejść pokolei, przerobić. Dziękuję, temat do zamknięcia.

Niestety, nie są to pojęcia łatwe i czytając definicję z wikipedii może być ciężko je opanować. Znalazłam kilka fajnych źródeł, z których można skorzystać.

### Materiały do nauki algorytmów po polsku

Zwróćcie uwagę, że nie jest wymagana umiejętność programowania, aby nauczyć się algorytmiki. Możecie te problemy rozwiązywać nawet układając klocki lego, jak długo pozwala wam to zrozumieć zasadę rozwiązywania problemów.

Tam, gdzie język programowania się przydaje to już implementacja (czyli napisanie kodu, zaprogramowanie) rozwiązania danego problemu.

► [**Algorytmy KhanAcademy**](https://pl.khanacademy.org/computing/computer-science/algorithms) – to darmowy, **interaktywny kurs podstaw algorytmiki** dostępny na platformie Khan Academy. Bardzo przyjemnie przedstawione zagadnienia z przykładami, dużą liczbą ilustracji, zadań do samodzielnego rozwiązania, dodatkowo kurs zawiera wyzwania w formie uzupełnianki kodu JavaScript (nie są konieczne do zrozumienia kolejnych lekcji). Rozwiązanie całości to zapewne od kilku do kilkunastu godzin nauki w zależności od indywidualnego tempa. Dzieląc sobie na 1-2 godzinę dziennie, ze spokojem zrobimy kurs w 2 tygodnie.


► [**Algorytmy.org**](http://www.algorytm.org/) – strona w całości poświęcona algorytmice. Znajdziemy na niej sporo tematów z tych co podałam powyżej. Były moją stroną startową przez semestr, gdy na uczelni trwały algorytmy i struktury danych.

Od wprowadzenia teorytycznego jako [kurs algorymiki](http://www.algorytm.org/kurs-algorytmiki/), przez [struktury danych](http://www.algorytm.org/struktury-danych/) i [algorytmy](http://www.algorytm.org/algorytmy/) z przykładami. Jest z czego wybierać. Co najważniejsze na stronie dostępne są do pobrania przykładowe implementacje (głównie w C++).


**Zadanie:**

**Na podstawie pokazanego przykładu stwórz inny przykład wymyślony przez Ciebie**

**Kolejność czynności, prowadzących do rozwiązania problemu za pomocą komputera**

**Etapy konstruowania algorytm** :

1) sformułowanie zadania – ustalamy jaki problem ma rozwiązywać algorytm

2) określenie danych wejściowych – ich typu ( w typie określamy, czy dane są liczbami rzeczywistymi, całkowitymi, czy znakami, czy też innego typu)

3) określenie wyniku oraz sposobu jego prezentacji

4) ustalenie metody wykonania zadania (może być kilka metod na rozwiązanie –

wybieramy tą, która według nas jest najlepsza)

5) zapisanie algorytmu za pomocą wybranej metody (z punktu 4 )

6) Analiza poprawności rozwiązania

7) Testowanie rozwiązania dla różnych danych (algorytm musi być uniwersalny, aby

służyć do rozwiązywania zadań dla różnych danych wejściowych)

8) Ocena skuteczności algorytmu ( praktyczna ocena algorytmu : np. szybkości,

skomplikowania)

**Do najczęściej używanych sposobów zapisu algorytmu należą :**

1) lista kroków

2) pseudojęzyk (pseudokod)

3) graficzna prezentacja za pomocą schematu blokowego

4) zapis w danym języku programowania

**Zadanie : znaleźć średnią arytmetyczną dwóch liczb rzeczywistych**

Lista kroków charakteryzuje się tym, że każdy wiersz opisujący pojedynczy krok realizowanej czynności jest numerowany.

1) pobierz pierwszą liczbę

2) pobierz drugą liczbę

3) dodaj liczby do siebie

4) wynik dodawania podziel przez 2

5) wyświetl otrzymaną wartość

6) zakończ

Pseudojęzyk jest metodą pośrednią między zapisem za pomocą listy kroków a zapisem w języku programowania.

- początek

- wprowadzenie x i y rzeczywistych

- wykonanie działania (x+y)/2

- pisz wynik

- koniec

Ten problem zapisany w postaci programu w języku Turbo Pascal

Program Srednia;

Var x, y : Real;

Begin

Readln (x);

Readln (y);

Writeln (‘Średnia arytmetyczna wprowadzonych liczb wynosi : ‘ (x+y)/2 :7:2);

End.

Przedstawiony algorytm liczenia średniej jest wykonywany zawsze w tej samej kolejności,

niezależnie od wartości danych wejściowych.

