

NOTATKI I UZUPEŁNIENIA DO WYKŁADU 1

8 października 2016

PROBLEM ALGORYTMICZNY

1. Zrozumienie problemu
2. Określenie zestawu **danych** wejściowych oraz **wyniku**
3. Konstrukcja **algorytmu**: ciąg jasno określonych kroków koniecznych do rozwiązania problemu
4. Zapis algorytmu w **pseudokodzie**: oddaje strukturę algorytmu aby łatwo zaimplementować go w **dowolnym języku programowania**
5. Określenie złożoności czasowej algorytmu
6. Weryfikacja poprawności algorytmu

Struktury danych

Sposób przechowywania informacji
w komputerze

Najprostsze struktury danych

Zmienna

miejsce, w którym można przechowywać dane; zmiennej nadajemy **nazwę**

zmienna odpowiadająca liczbie

zmienna odpowiadająca wartości logicznej

(**true** lub **false**)

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX

Instrukcja przypisania

zmienna = wyrażenie

powoduje, że wyrażenie po prawej stronie = ulega obliczeniu i jego wartość jest przypisywana zmiennej

Wyrażenie – liczba, dowolna zmienna lub operacje arytmetyczne lub logiczne

Operatory arytmetyczne: $+$ $-$ $*$ $/$ mod div

Operatory logiczne: **or** **and** **not**

Znak równości: $==$

Znak różności: \neq

zmiennej X przypisz liczbę dwa

$$X = 2$$

zwiększ wartość X o cztery

$$X = X + 4$$

wyrażenia logiczne

$$(x > 5) \textbf{ or } (x \bmod 2 == 0)$$

$$(x > 5) \textbf{ and } (x \bmod 2 == 0)$$

Wielokrotne przypisywanie w formie

$$i = j = \text{wyrażenie}$$

oznacza przypisanie zmiennym i oraz j wartości wyrażenia stojącego po prawej stronie

Instrukcja **return**

zwraca wartość i przerywa wykonywanie algorytmu, z wyjątkiem
....

Przykład 1. Podnoszenie liczby do kwadratu

KWADRAT(x)

```
1   $y = x * x$ 
2  return  $y$ 
```

Iteracja ograniczona **for**

for zmienna sterująca = wartość początkowa
 to wartość końcowa **do** instrukcje

for zmienna sterująca = wartość początkowa
 downto wartość końcowa **do** instrukcje

Wartość początkowa i wartość końcowa: liczby, zmienne lub wyrażenia arytmetyczne

```
for  $i = M$  to  $N$ 
    do  $A$ 
```

Instrukcja A zostanie powtórzona $N - M + 1$ razy a wartość zmiennej sterującej i po wyjściu z iteracji będzie wynosiła $N + 1$

```
for  $i = N$  downto  $M$ 
    do  $A$ 
```

Instrukcja A zostanie powtórzona $N - M + 1$ razy a wartość zmiennej sterującej i po wyjściu z iteracji będzie wynosiła $M - 1$

Przykład 2. Rozważmy dwa ciągi instrukcji:

```

1   $s = 0$ 
2  for  $i = 1$  to  $n$ 
3    do  $s = s + 1/i$ 

```

```

1   $s = 0$ 
2  for  $i = n$  downto  $1$ 
3    do  $s = s + 1/i$ 

```

Czy w wyniku działania tych dwóch ciągów instrukcji otrzymamy ten sam wynik?

Przykład 3. Algorytm obliczający wartości wyrażenia a^n , gdzie $a \neq 0$, natomiast n jest nieujemną liczbą całkowitą.

Dane wejściowe: liczba $a \neq 0$

liczba całkowita $n \geq 0$

Wynik: wartość a^n

Konstrukcja algorytmu:

$$a^n = a \cdot a \cdot \dots \cdot a$$

($n - 1$ operacji mnożenia)

POTEŃGA(a, n)

```

1   $b = a$ 
2  for  $k = 1$  to  $n - 1$ 
3    do  $b = b * a$ 
4  return  $b$ 

```

POTEŃGA(a, n)

```

1   $b = 1$ 
2  for  $k = 1$  to  $n$ 
3    do  $b = b * a$ 
4  return  $b$ 

```

```

POTĘGA ( $a, n$ )
   $b = 1$ 
  for  $k = n$  downto 1
    do  $b = b * a$ 
  return  $b$ 

```

Iteracja warunkowa while

```

while  $Q$ 
  do  $A$ 

```

Obliczana jest wartość logiczna warunku Q . Jeżeli warunek ma wartość **true**, wykonywana jest instrukcja A . Proces ten jest powtarzany dopóki warunek nie przyjmie wartości **false**

Przykład 4. Zapisać algorytm z Przykładu 3 (wersja **downto**) wykorzystując iterację warunkową **while**

```

POTĘGA ( $a, n$ )
1   $b = 1$ 
2   $k = n$ 
3  while  $k \neq 0$ 
4      do  $b = b * a$ 
5           $k = k - 1$ 
6  return  $b$ 

```

Iteracja warunkowa repeat

```

repeat
   $A$ 
until  $Q$ 

```

Wykonywana jest instrukcja A . Obliczana jest wartość logiczna warunku Q . Proces jest powtarzany dopóki warunek Q nie przyjmie wartości **true**

Przykład 5. Zapisać algorytm z Przykładu 4 wykorzystując iterację warunkową **repeat**

POTEŃGA (a, n)

```

1   $b = 1$ 
2   $k = n$ 
3  repeat
4     $b = b * a$ 
5     $k = k - 1$ 
6  until  $k == 0$ 
7  return  $b$ 
```

$n = 0$?

Przykład 6. Podaj ciąg instrukcji wczytujący liczby różne od zera i zwracający ich sumę (nie wiemy ile liczb jest do wczytania).

Wykorzystać instrukcję
czytaj (*zmienna*)

```

1   $s = 0$ 
2  repeat czytaj( $a$ )
3     $s = s + a$ 
4  until  $a == 0$ 
5  return  $s$ 
```

Przykład 7. Algorytm, który dla zadanej liczby naturalnej wypisuje jej dziesiętną reprezentację w odwrotnym porządku (np. dla $n = 173$ algorytm ma wypisać liczbę 371, a dla $n = 1000$ wynikiem będzie 0001)

Wykorzystać instrukcję
pisz (*wyrażenie*)

```

ODW( $n$ )
  while  $n \neq 0$ 
    do pisz ( $n \bmod 10$ )
       $n = n \operatorname{div} 10$ 

```

```

ODW( $n$ )
  repeat
    pisz ( $n \bmod 10$ )
     $n = n \operatorname{div} 10$ 
  until  $n == 0$ 

```

Wybór warunkowy

```

if  $Q$ 
  then  $A$ 

```

Jeśli warunek logiczny Q ma wartość **true**, wykonana będzie instrukcja A ; w przeciwnym razie (Q ma wartość **false**) przechodzimy do następnej instrukcji

```

if  $Q$ 
  then  $A$ 
  else  $B$ 

```

Jeśli warunek Q ma wartość **true**, wykonana będzie instrukcja A ; w przeciwnym razie będzie wykonana instrukcja B

Przykład 8. Algorytm **naiwny** obliczający największy wspólny dzielnik $\text{NWD}(a, b)$, gdzie a i b są dodatnimi liczbami całkowitymi

```

NWD( $a, b$ )
1  if  $a > b$ 
2    then  $k = b$ 
3    else  $k = a$ 
4  while ( $a \bmod k \neq 0$ ) or ( $b \bmod k \neq 0$ )
5    do  $k = k - 1$ 
6  return  $k$ 

```

Przykład 9. NWD – Euklides

1. Oznacz przez M jedną, a przez m drugą z liczb a , b .
2. Oblicz resztę r z dzielenia M przez m .
3. Jeśli $r = 0$, to $\text{NWD}(a, b) = m$;
w przeciwnym razie, w miejsce M podstaw m , a w miejsce m podstaw r i przejdź do kroku 2.

Tablica

Wiele zmiennych można połączyć w **strukturę danych** zwaną **tablicą** i oznaczyć wszystkie te zmienne nazwą tablicy z dodatkiem jednego lub więcej **indeksów**, przebiegających wartości całkowite. Indeksami mogą być *liczba*, *zmienna* lub *wyrażenie*

Tablice jednowymiarowe – wektory

Opis tablicy

nazwa[indeks początkowy .. indeks końcowy]

$A[1..35]$, $B[1..n]$, $C[k..k + j]$

Dostęp do elementu wektora

nazwa[indeks]

$A[24]$, $B[n \text{ div } 2]$, $C[m]$ ($k \leq m \leq k + j$)

Instrukcja przypisania

nazwa[indeks] = wyrażenie

$B[j] = 3 * A[12] + C[k + j]$

Przykład 10. Wpisanie do tablicy $A[1..n]$ kolejno liczb $1, 2, \dots, n$

WPISYWANIE (A, n)

```

for  $i = 1$  to  $n$ 
  do  $A[i] = i$ 

```

Przykład 11. Algorytm wyznaczający największą wartość wektora $A[1..n]$ oraz wszystkie pozycje (wartości indeksów), na których się ona znajduje - wersja **naiwna**

MAX-POZ (A, n)

```

1   $max = A[1]$ 
2  for  $i = 2$  to  $n$ 
3      do if  $A[i] > max$ 
4          then  $max = A[i]$ 
5  pisz ( $max$ )
6  for  $i = 1$  to  $n$ 
7      do if  $A[i] == max$ 
8          then pisz ( $i$ )

```

Tablice dwuwymiarowe – macierze

Odwołanie do elementu macierzy uzyskuje się używając **pary uporządkowanej indeksów** (pierwszy indeks można interpretować jako numer wiersza, a drugi jako numer kolumny macierzy)

Przykład 12. Algorytm wpisujący do każdego elementu tablicy $A[1..m, 1..n]$ sumę numerów jego wiersza i kolumny

SUMY (A, m, n)

```

1  for  $i = 1$  to  $m$ 
2      do for  $j = 1$  to  $n$ 
3          do  $A[i, j] = i + j$ 

```

Pojęcie procedury

Deklaracja procedury

- nagłówek procedury
- treść procedury

Nagłówek procedury: **procedure** $P(WF); S$

P – nazwa procedury

WF – wykaz parametrów formalnych

S – specyfikacja parametrów formalnych

Parametr formalny

Jest obiektem występującym w treści procedury. Nie odpowiada mu bezpośrednio żaden obiekt algorytmu; dopiero w chwili użycia procedury, każdemu parametrowi formalnemu zostaje przyporządkowany parametr **aktualny**

Rodzaj parametru aktualnego musi być zgodny ze specyfikacją odpowiadającego mu parametru formalnego

Treść procedury

ma charakter instrukcji złożonej; mogą w niej występować obiekty dwojakiego rodzaju

- parametry formalne
- *obiekty lokalne* – obiekty wyspecyfikowane w treści procedury

Wywołanie procedury w algorytmie

$P(WA)$

P – nazwa procedury

WA – wykaz parametrów *aktualnych* zgodny
co do liczby i specyfikacji z wykazem
parametrów formalnych w deklaracji
procedury P

Przykład 13. Deklaracja procedury wyznaczającej największy
element tablicy $A[1..n]$

```

procedure MAX ( $A, n, w$ )
1      tablica rzeczywista  $A$ 
2      całkowite  $n$ 
3      rzeczywiste  $w$ 

4  begin
5      całkowite  $j$ 
6       $w = A[1]$ 
7      for  $j = 2$  to  $n$ 
8          do if  $A[j] > w$ 
9              then  $w = A[j]$ 
10     end

```

Zapis skrótowy

```

procedure MAX ( $A, n, w$ )
1       $w = A[1]$ 
2      for  $j = 2$  to  $n$ 
3          do if  $A[j] > w$ 
4              then  $w = A[j]$ 

```