

# Algorithm2e

Paweł Magiera

17 listopada 2019

# Plan prezentacji

- 1 Podstawowe informacje
  - Użycie pakietu
  - Przykład
- 2 Praca z pakietem
  - Komenda If
  - Switch/Case
  - While/For/For Each
  - Własne komendy
  - Komentarze
  - Zmiana stylu
- 3 Zadania

## Pakiet Algorithm2e

Algorithm2e to pakiet do  $\text{\LaTeX}$ , który dodaje funkcję łatwego wyświetlania algorytmów w dokumencie. Może być przydatny w pisaniu dokumentacji do programów komputerowych lub dokumentów matematycznych.

Aby móc korzystać z pakietu musimy wpisać w preambule dokumentu:

```
\usepackage[opcje]{algorithm2e}
```

Algorytmy piszemy w bloku:

```
\begin{algorithm}
```

```
\end{algorithm}
```

# Przykład użycia

```
\begin{algorithm}[H]
  \SetAlgoLined
  \KwData{this text}
  \KwResult{how to write algorithm with \LaTeX2e }
  \BlankLine
initialization\;
  \While{not at end of this document}{
    read current\;
    \eIf{understand}{
      go to next section\;
      current section becomes this one\;
    }{
      go back to the beginning of current section\;
    }
  }
}
```

# Przykład

**Data:** this text

**Result:** how to write algorithm with  $\text{\LaTeX}$ 2e  
initialization;

```
while not at end of this document do  
|   read current;  
|   if understand then  
|   |   go to next section;  
|   |   current section becomes this one;  
|   else  
|   |   go back to the beginning of current section;  
|   end  
end
```

# Instrukcje warunkowe

## If

```
\If{warunek}  
{ciało ifa}
```

```
if warunek then  
|   ciało ifa  
end
```

## If bez "end"

```
\uIf{warunek}  
{ciało ifa}
```

```
if warunek then  
|   ciało ifa
```

## If w jednej linijce

```
\lIf{warunek}{ciało ifa}
```

```
if warunek then ciało ifa;
```

Domyślnie polecenia kończą się słowem kluczowym "end" jednak można tego uniknąć dodając "u" przed poleceniem. Aby wyświetlić polecenie w jednej linijce należy analogicznie napisać "l".

# Instrukcje warunkowe

## uElse

```
\uIf{warunek}  
{ciało ifa}  
\uElse  
{ciało else}
```

```
if warunek then  
|   ciało ifa  
else  
|   ciało else
```

## Else

```
\ElseIf{warunek}  
{ciało else ifa}
```

```
else if warunek then  
|   ciało else ifa  
end
```



# Instrukcje warunkowe

## Switch

```
\Switch{warunek} {  
  \Case{przypadek1}{ciało przypadku}  
  \Case{przypadek2}{ciało przypadku}  
}
```

```
switch warunek do  
|   case przypadek1 do  
|     |   ciało przypadku  
|   end  
|   case przypadek2 do  
|     |   ciało przypadku  
|   end  
end
```

# Pętle

## While

```
\While{warunek}  
{ciało while}
```

```
while warunek do  
|   ciało while  
end
```

## For

```
\For{$i\leftarrow 0$  
\KwTo $10$}{print $i$}
```

```
for  $i \leftarrow 0$  to 10 do  
|   print  $i$   
end
```

## For Each

```
\ForEach{warunek}  
{ciało foreach}
```

```
foreach warunek do  
|   ciało foreach  
end
```

# Własne komendy

## Deklaracja komendy

```
\SetKwProg{Fn}{Funkcja}{ :}{koniec}}
```

## Sposób użycia

```
\Fn{mojaFunkcja}{\Return{0}}
```

Funkcja *mojaFunkcja* :  
| return 0  
koniec

# Komentarze

## komentarze

```
\tcc{komentarz}
```

```
\tcc*[r]{komentarz wyrównany do prawej}
```

```
\tcc*[l]{komentarz wyrównany do lewej}
```

```
/* komentarz                                     */  
;          /* komentarz wyrównany do prawej */  
; /* komentarz wyrównany do lewej           */
```

# Komentarze

## komentarze

```
\tcp*[l]{will be used to compute x}
```

```
\tcp*{will be used to compute x}
```

```
; // will be used to compute x
```

```
;           // will be used to compute x
```

# Zmiana stylu

Aby zmienić kolor np. tekstu w jednej linii, lub kolor komentarzy, wymagany jest pakiet xcolor.

```
\usepackage{xcolor}
```

# Kolorowanie algorytmu

## Zmiana stylu komentarzy

```
\newcommand\mycommfont[1]{  
\footnotesize\textcolor{blue}{#1}  
}  
\SetCommentSty{mycommfont}  
  
; // will be used to compute x
```

# Kolorowanie algorytmu

## Kolorowanie części algorytmu

```
\Switch{warunek}{  
  \Case{\textcolor{red}{przypadek}}  
  {\textcolor{gray}{ciało przypadku}}}
```

```
switch warunek do  
|   case przypadek do  
|   |   ciało przypadku  
|   end  
end
```



# Styl wyświetlania

Auto numerowanie

`\LinesNumbered`

Numerowanie ręczne

`\n1 \n2 \n3 ...`

Linie proste

`\SetAlgoLined`

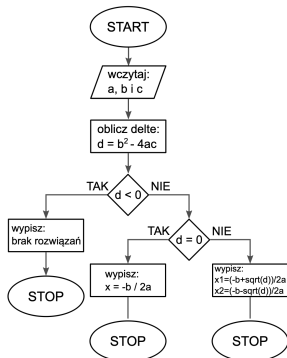
Linie z wcięciem

`\SetAlgoVlined`

```
1 switch warunek do
2   | case przypadek do
3   |   | ciało przypadku
```

# Zadania

- 1 Zapisz poniższy algorytm za pomocą pakietu algorithm2e:



# Zadania

- 2 Stwórz dowolną własną komendę.
- 3 Znajdź dowolny algorytm (ok. 15 linijek) i zapisz go za pomocą pakietu algorithm2e. Powinien on zawierać:
  - a Informacje na temat danych wejściowych i wyjściowych (Data, Result)
  - b Przynajmniej jedną pętlę
  - c Komentarze

# Zadania

- ❶ Zapisz poniższy algorytm tak, aby wyglądał identycznie jak poniżej:

Data: A bitmap  $Im$  of size  $w \times l$

Result: A partition of the bitmap

```
1  for  $i \leftarrow 2$  to  $l$  do
2       $j \leftarrow 2$  to  $w$  left  $\leftarrow$  FindCompress( $Im[i, j - 1]$ )
3       $up \leftarrow$  FindCompress( $Im[i - 1, j]$ )
4       $this \leftarrow$  FindCompress( $Im[i, j]$ )
5      if left compatible with this then
6          if left < this then Union(left, this);
7          else Union(this, left);
8      if up compatible with this then
9          if up < this then Union(up, this);
10         // this is put under up to keep tree as flat as possible
11         else Union(this, up);
12         // this linked to up
13     foreach element  $e$  of the line  $i$  do FindCompress(p);
```