

PAKIET ALGORITHM

Jakub Karmański

POLSL

18.11.2019

Algorithm 1: Zad 1

pobierz a,b,c;

oblicz delte -> $\text{delta} = b^2 - 4ac$

if *delta* < 0 **then**

| brak rozwiązań;

else

| **if** *delta* = 0 **then**

| | $x = -b/2a$;

| **else**

| | $x1 = -b + \text{sqrt}(\text{delta})/2a$;

| | $x2 = -b - \text{sqrt}(\text{delta})/2a$;

| **end**

end

Funkcja *mojafunkcja*(*i: int*) : *int* is

| return 0;

end

def *mojafunkcja*(*i: int*):

| return 0;

Data: this text

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

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

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

```