

Algorithmic Thinking

Explaining Algorithms

G. Futschek

2009

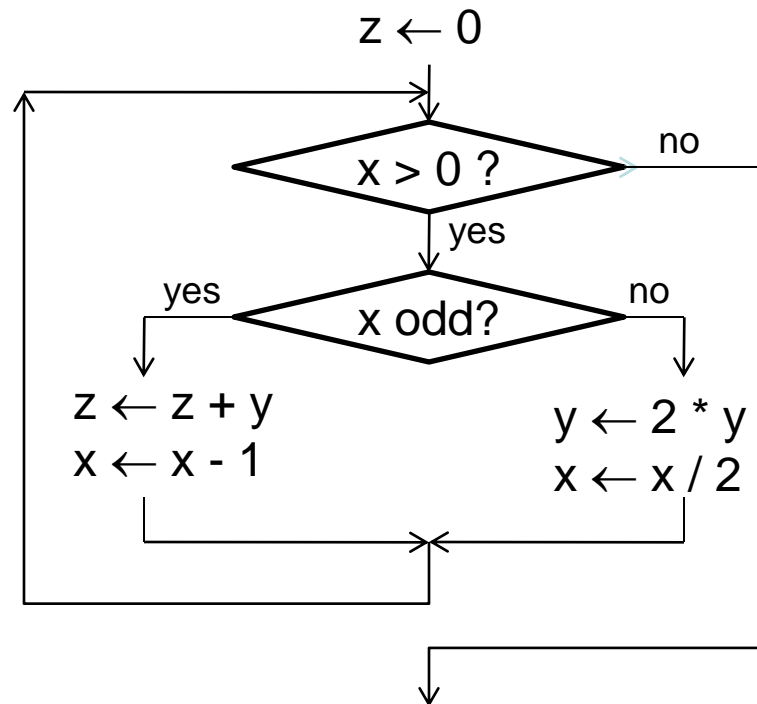
Experiment 1

```
n ← length( A )  
do  
  swapped ← false  
  n ← n - 1  
  for i ← 0 to n - 1 do  
    if A[ i ] > A[ i + 1 ] then swap( A[ i ], A[ i + 1 ] )  
    swapped ← true  
  end if  
  end for  
while swapped
```

Explain this algorithm!

Experiment 2

Mystery($\downarrow x$, $\downarrow y$, $\uparrow z$):



From Mössenböck:
Sprechen Sie Java

Explain this algorithm!

Experiment 3

```
L ← Empty list that will at end contain the result
S ← Set of all nodes with no incoming edges
while S is non-empty do
    remove a node n from S
    insert n into L
    for each node m with an edge e from n to m do
        remove edge e from the graph
        if m has no other incoming edges then
            insert m into S
if graph has edges then
    output error message (graph has at least one cycle)
else output L
```

from Wikipedia

Explain this algorithm!

What means “explain“?

- What occurs step by step?

or

- What is changing?

or...

What means “explain“ really...

- What are the inputs, what are the outputs?
- What is the aim of the algorithm?
(postcondition)
- What are correct input data (precondition)?
- What is kept invariant?
- Why terminates the algorithm?

Essence of Explaining

explain,
why
the algorithm
achieves always
the correct result

Reference: G. Futschek: Explaining and Understanding LOGO Programs, a Discipline of Learning Computer Programming, Proceedings of Eurologo 2005.

Documentation of Programs

to remove :ch :w

; precondition: ch is a character, w is a word with n characters

; postcondition: outputs word w from which the first occurrence of ch was removed

; runtime $O(n)$, algorithm: linear search

if empty? :w [op :w]

; assert: w has at least one character

if :ch = first :w [op bf :w]

op word (first :w) (remove :ch bf :w)

end