University of Strathclyde Glasgow

Computational thinking

W. H. Bell

September 4, 2023

Abstract

Processes of computational thinking are introduced, including abstraction, decomposition and pattern recognition. Within the presented examples, a flowchart and pseudocode are used to represent an algorithm or computer program.

1 Introduction

An algorithm is a set of instructions that is used to complete a process or solve a problem. A computer program is the implementation of the algorithm. To implement a computer program, a process must be described as a set of steps and options that might be taken.

A software developer decides which details of a system need to be expressed in an algorithm, which is known as abstraction. The software developer then breaks the process down into component steps, which is known as decomposition. During decomposition, a developer looks for repeated patterns of functionality that will become subroutines or functions within the final implementation.

An algorithm can be expressed as structured English, a flowchart or pseudocode. Once a developer is familiar with a computer programming language and as long as the logic of the algorithm is relatively straightforward, a developer may only document the algorithm using structured English before it is implemented. If the developer spends no time with structured English or another method of documenting decomposition, they risk not completely understanding the algorithm. This can cause a developer to be confused as to how a computer program should be constructed, wasting precious development time.

Flowcharts can be used to precisely document more complex algorithms. They require more time to implement, but can be clearly understood by other developers. Flowcharts are constructed against a set of rules that govern how the diagrams are created. Flowcharts are often not used for routine software development, since they require a greater time investment.

Pseudocode is structured English that is written using a syntax that is more similar to the final computer program implementation, but is not a computer program itself. There is no fixed standard for

pseudocode. Developers may use syntax for documenting pseudocode that is similar to one specific final computer language. The pseudocode that is given in this document is generic and simplistic to introduce the concept.

In summary, a computer program is constructed by:

- Understanding the user requirement(s).
- Thinking through the steps needed to achieve the requirements.
- Thinking about variables or options that are needed.
- Identifying repeated patterns that will become subroutines or functions.
- Documenting the decomposition using structured English.
- Optionally, expressing steps as a flowchart or pseudocode.
- Implementing the algorithm in a computer programming language.
- Testing that the implemented logic fulfils the user requirements.

2 Brushing teeth

Exercise

Design an algorithm that describes the process of a person brushing their teeth with a toothbrush and toothpaste. The algorithm should start with a person who needs to clean their teeth and has access to a dry toothbrush and new tube of toothpaste. The program should end with the person having successfully cleaned their teeth.

Solution

Steps

- Remove top from toothpaste.
- Squeeze toothpaste until correct amount is on toothbrush.
- Replace top on toothpaste.
- Start brushing teeth.
- Stop brushing teeth.
- Turn on cold tap.
- Wash toothbrush under tap.
- Turn off tap.

Options

- The amount of toothpaste.
- The time taken brushing teeth.

Flowchart

A flowchart description of the process of brushing teeth is given in Figure 1. The flowchart comprises a series of processes and decisions.

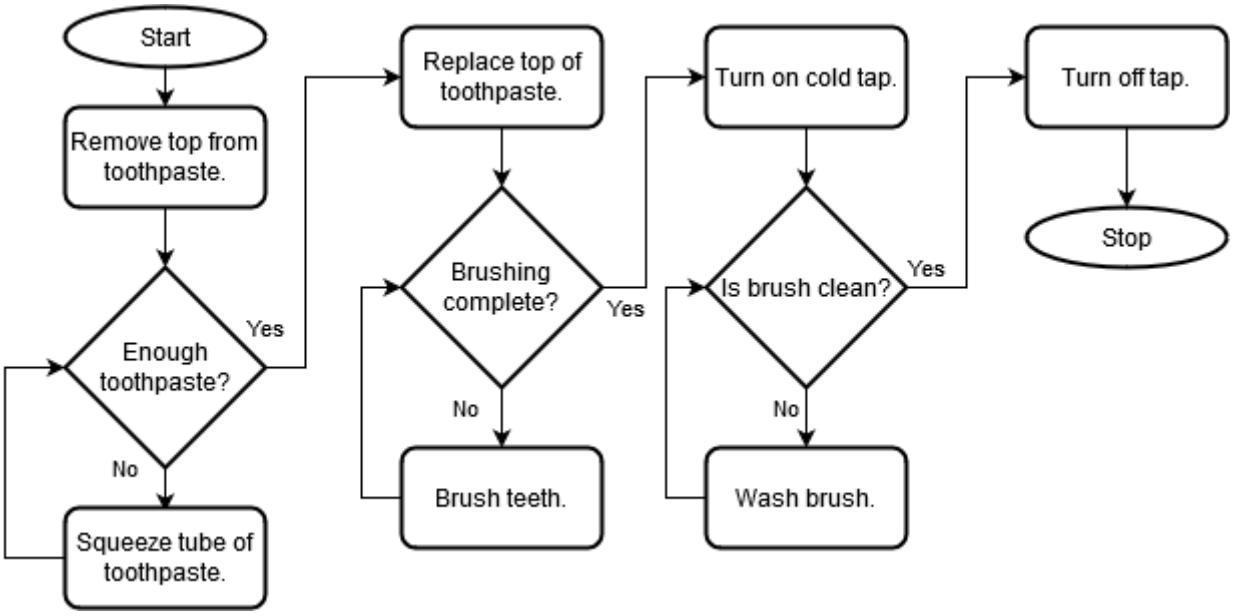

Figure 1: A flowchart to describe the process of brushing teeth.

Pseudocode

The algorithm is described in Listing 1, using the pseudocode rules that are defined in the “Cambridge International AS & A Level Computer Science Pseudocode Guide for Teachers”. Within this standard, the character “←” is used to represent the assignment of a value to a variable. All keywords are given in upper case characters.

The algorithm starts by asking the user to input the amount of toothpaste needed and the brushing time needed, storing the values in the variables `ToothpasteNeeded` and `BrushingTimeNeeded`, respectively. The process of removing the top, squeezing the tube, replacing the top, brushing teeth, turning on the tap and turning off the tap are not defined in the presented pseudocode. Therefore, they are shown as function calls that should be defined elsewhere. The algorithm continues to squeeze the tube until

enough toothpaste is present. Likewise, the algorithm continues to brush teeth until the brushing time is complete. When the brush is washed, another function is called to check the state of the brush. The return value from `CheckBrushState()` is assigned to `BrushIsClean`. This corresponds to the person looking at the toothbrush and verifying if the brush is clean or not.

Listing 1: Pseudocode to describe the process of brushing teeth.

```
INPUT ToothpasteNeeded
INPUT BrushingTimeNeeded
RemoveTop()
Toothpaste ← 0
WHILE Toothpaste < ToothpasteNeeded DO
    SqueezeTube()
    Toothpaste ← Toothpaste + 10
ENDWHILE
ReplaceTop()
BrushingTime ← 0
WHILE BrushingTime < BrushingTimeNeeded DO
    BrushTeeth()
    BrushingTime ← BrushingTime + 10
ENDWHILE
TurnOnTap()
BrushIsClean ← FALSE
WHILE NOT BrushClean DO
    WashBrush()
    BrushIsClean ← CheckBrushState()
ENDWHILE
TurnOffTap()
```

3 Making tea

Exercise

Design a computer program that describes the process of making a cup of tea with hot water and milk. The program should start with an empty kettle and a cup. The program should end with a cup that contains hot tea that is ready to drink.

Follow the example that is given in Section 2 and create a solution that includes steps, options, a flowchart and pseudocode. Use https://app.diagrams.net/ to create the flowchart.