CS3000: Algorithms & Data — Summer '25 — Laney Strange

Pseudocode Guide

Pseudocode is a kind of structured English for describing algorithms. It allows the designer to focus on the logic of the algorithm without being distracted by details of language syntax. At the same time, the pseudocode needs to be complete. It describes the entire logic of the algorithm so that implementation becomes a rote mechanical task of translating line by line into source code.

In CS3000, we'll often ask you to generate pseudocode to describe your algorithmic solution to a problem. We expect your pseudocode to follow the guidelines listed here.

Overview

Steps to get started

- Download clrspseudo.sty from the course website
- Save it in the same location where you have your source .tex file
- To use the package, include the following line in your source file: usepackage{clrspseudo}

CLRS pseudocode conventions

- Indentation denotes a code block
- = denotes assignment and we say and type "gets" (e.g., key = A[i])
- Variables are local to the given procedure
- Arrays are the basic data struture (similar to lists in Python)
- Arrays are indexed from 1, not 0
- \bullet Control structures such as while, for, and if behave as you would expect in Python, Java, or C/C++

Producing CLRS pseudocode in IAT_{FX} .

- Use \begin{codebox} to start a pseudocode block and \end{codebox} to end it.
- Use \Procname{} to name the procedure for your algorithm. It takes as a parameter the procedure name and parameters, typically all in math mode. All together it looks like this: \Procname{\$\proc{My-Procedure}(Param1, Param2)\$}.
- Use \li to begin a new numbered line in your algorithm.
- Use \id{} for an identifier (variable name).

- The following commands simply produce their corresponding keywords, typeset in boldface: \While, \For, \If, \ElseIf, \Else, \To, \By, \Return, \Error.
- Use \const{} for constants such as TRUE, FALSE, and NIL.
- To correctly indent a loop, use \Do and \End along with \For or \While. They do not appear on their own after an \li (see below for an example).
- To correctly indent a conditional structure, use \Then and \End along with \If, \ElseIf, and \Else (see below for an example).

Example with an array and a loop

Suppose you want to typeset the following pseudocode:

```
SUM-ARRAY(A, n)

1 sum = 0

2 for i = 1 to n

3 sum = sum + A[i]

4 return sum
```

You would type the following into $\square T_EX$:

```
\begin{codebox}
\Procname{$\proc{Sum-Array}(A, n)$}
\li $\id{sum} \gets 0$
\li \For $i \gets 1 \To n$
\Do
\li $\id{sum} \gets \id{sum} + \id{A[i]}$
\End
\li \Return $\id{sum}$
\end{codebox}
```

Example with a single parameter, a conditional, and use of constants

Suppose you want to typeset the following pseudocode:

ISEVEN(n)1 if $(n \mod 2) == 0$ 2 return TRUE 3 else 4 return FALSE

You would type the following into IAT_EX :

```
\begin{codebox}
\Procname{$\proc{IsEven}(n)$} \\
\li \If $(n \mod 2) == 0$
\Then
\li \Return \const{True}
\li \Else
\li \Return \const{False}
\End
\end{codebox}
```