Pseudocode Best Practices

Rather than continuing to **tell** you what you should and shouldn't do in pseudocode, we thought we'd try to let you determine for yourself (based on reading **our** pseudocode!) what is and isn't helpful in pseudocode.

Another helpful general hint: if you're looking for the general style we like in pseudocode, try looking at algorithms we include in assignment instructions and/or in sample solutions to assignment and worksheet problems. The point in pseudocode in general is to be understandable. That **is** our goal when we write algorithms for assignment instructions and sample solutions (hard as you may sometimes find that to believe...).

Pseudocode comparisons

Below we'll provide different ways to express the same algorithm, ranging from actual code to a couple of sentences. Read these over and try to understand what they do.

Type 1: Some real code

I do a lot work in MATLAB. So, ~~just to torture you~~ just for fun, here's an algorithm I implemented in MATLAB:

```
function Y = SomethingMysterious(X)

Y = {};
while length(X)
    w = X(1);
    c_w = 1;
    inds = [1];
    for i=2:length(X)
        if strcmp(X(i), w)
            c_w = c_w+1;
            inds = [inds i];
        end
    end
    Y{end+1} = {w, c_w};
    X(inds) = [];
end

end
```

1. How easy do you find it to **understand** what the algorithm in `SomethingMysterious` is doing?

2. If you needed to **analyze** `SomethingMysterious` (e.g., if we put this on an assignment and asked you to determine the running time), how easy would you find that to do?

Type 2: Slightly More Code-like pseudocode

Below I've taken `SomethingMysterious` and converted it to pseudocode. While pseudocode-writing style varies, this is a bit more detailed than I usually write.

```
PROCEDURE SomethingMysterious_v2(X):
    Y = [ ]
    while length(X) > 0:
        let w be the first element of X
        initialize the count c_w to 1
        initialize inds (indices to delete later) to the single-element list [1]
        for i=2:length(X)      // assume 1-based indexing
            if X[i] = w:
                c_w = c_w + 1
                append i to inds
        append {w, c_w} to Y
        delete from X all values at the indices stored in the array inds
```

1. How easy do you find it to **understand** what the algorithm in `SomethingMysterious_v2` is doing?

2. If you needed to **analyze** `SomethingMysterious_v2` (e.g., if we put this on an assignment and asked you to determine the running time), how easy would you find that to do?

Type 3: Slightly More English-like pseudocode

This is a more high-level version of `SomethingMysterious`, which is more aligned with my personal preference for pseudocode-writing (though, again, there is some leeway for individual style/preferences here):

```
PROCEDURE SomethingMysterious_v3(X):
    Y = [ ]
    While X is not empty:
        Let w be the first element of X
        Count the number of occurrences of w in W (by scanning W from start to
           end) and call this result c_w
        Append w:c_w to Y
        Delete all occurrences of w from X
    return Y
```

1. Hopefully you've realized by now (or from one of the previous examples) that you have **already** been required to understand and analyze this function in quiz and assignment 1 (though we used more informative function and variable names in that case).

   Do you remember how easy or difficult it was to understand the algorithm when you first saw it? How do you think your experience might have varied if we had presented the algorithm in the style of `SomethingMysterious` or `SomethingMysterious_v2` instead?

Type 4: Much More English-like pseudocode

Another alternative to actual pseudocode – which sometimes doesn't provide enough detail but is sometimes perfectly adequate – is to just describe your algorithm in plain English sentences. For `SomethingMysterious`, that might look like:

> Start with an empty bag of words. For each word in W, count the number of occurrences and append the word and its count (as a tuple) to the bag of words. Then delete all the occurrences of that word from W, and continue until W is empty.

1. Do you think this would have provided enough detail to help you understand and analyze the algorithm in quiz/assignment 1? Why or why not?

Pseudocode Best Practices

Discuss with your group what **you** think makes for "good" or "bad" pseudocode. Your TAs can provide guidance here too, and we'll provide some of our own thoughts in the sample solution for this exercise. But try to think of what you've learned from this exercise and from reading and writing pseudocode!

Common questions we get (on Piazza, etc.) are: can I use a (some particular well-known function) without actually implementing it? Can I use (some particular data structure) without describing it in detail?

Pretend you're a CPSC 320 TA, and a student asked you this sort of question. How do you think you might answer them (given that, as the student's TA, you are likely to have to **read** their pseudocode)?