Pseudocode

In this class we use pseudo code to express our algorithms instead of using an actual compiled language such as java, python, or c++. Pseudo code allows us to avoid language specific implementation. It also us to express our algorithms with a few lines of high level descriptions and mathematical notations making the algorithm more concise and easier to read. There is no formal or designated way of writing pseudo code, but the problems on the exam will follow the (different) styles on the lectures.

Example: Min Sort

Pseudocode

<table>
<thead>
<tr>
<th>Algorithm 1: MinSort</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 for k ← 1 To n − 1 do</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11 end</td>
</tr>
</tbody>
</table>

Equivalent C-code

for(int k = 1; k ≤ n − 1; k++) {
    int min = A[k];
    int index = k;
    for(int j = k + 1; j ≤ n; j++){
        if(A[j] < min) {
            min = A[j];
            index = j;
        }
    }
    A[index] = A[j];
    index = j;
}

A few things to note

- Types: types do not have to be explicitly stated, if they are fairly obvious such as integer, decimal values and strings. Do not worry about implementation level problems associated with various types such as overflow. Remember pseudo code is suppose to high level

- Assignment: the ← is used in this class to represent an assignment operation as to avoid confusion with the = (equality) operation. Sometimes := is used for assignment as well.

- Sometimes we want our algorithms to report information back to the user. For our purpose, the print statement will suffice. If we print a variable, then the output is the value of the variable. If we print something enclosed in quotes, then the output consists of the quoted text. For example: print "Old value of x:" x
• Sometimes we also want our algorithms to be able to execute certain statements depending on whether some condition holds. The if ... then statement is the pseudocode version of this construction. When the statement if (condition) then (statement) executes, the program first checks to see if the (condition) is true, and, if it is, the program executes the (statement).

• High level description: Although not in the example above, sometimes there would instruction to use a different algorithm as a step in a given algorithm. Think of this as similar function call in a programming language. It is useful to know the basic algorithms taught in this class as they might be used as subroutines.

• Parameters: The parameters to a algorithm is usually explicitly given, unless obvious like in the example. Arrays and lists are usually represented as $A[1...n]$ (Array of size n) or $A_1...A_n$ (list of size n)

• To in for loops: A loop from $x$ to $y$ has both $x$ and $y$ as inclusive values. (Look at example) Here is another representation we will use in our graph algorithms.

  for $i:=1$ to $n-1$

• Indentations and curly braces: Both can be used in pseudo code. If you are not familiar with either try to look at the example to see the differences.

• Mathematical notation: Mathematical notations is often used to be concise. The notation in this class isn’t advanced but will cover set notation

### Set Notation Review

A set is just a group of items of ordered items. We often use set in place of actual data structure in pseudocode.

example

\[
A = \{a, b, c, d\} \\
B = \{a, c\}
\]

Membership: the $\in$ and $\notin$ are use to represent membership of an element within a set

example

\[
a \in A \\
f \notin A
\]

Subset: the $\subseteq$ means that every element of the first is also on the second set. The $\not\subseteq$ is the opposite.

example

\[
B \subseteq A \\
A \not\subseteq B \\
A \subseteq A
\]

Equivalence: the $=$ show equivalence. Two sets are equivalent if their elements are exactly the same

example

\[
A = A
\]

Cardinally: the $| |$ is used to show the number of elements in a set

example

\[
|A| = 4 \\
|B| = 2
\]

You are expected to know these concepts at the start of the class. If you have any questions ask us after class, on Piazza or at office hours.