6 Reading Assignment 1, Discrete Math, NYU


6.1 Comparison of orders of magnitude

We introduced the $O$, $\Omega$, $\Theta$-notation in class, please read the definitions again. We shall here make some additional examples.

Example 1. $1 + 2 + \cdots + n = \Theta(n^2)$. Proof: Let $\lfloor x \rfloor$ denote the largest integer smaller than $x$. Clearly,

\[ 1 + 2 + \cdots + n \geq \lfloor \frac{n}{2} \rfloor + \cdots + n \]
\[ \geq (\lfloor \frac{n}{2} \rfloor)^2 \sim \frac{n^2}{4}. \]

Therefore, $1 + 2 + \cdots + n = \Omega(n^2)$. On the other hand,

\[ 1 + 2 + \cdots + n < n + \cdots + n = n^2, \]

hence $1 + 2 + \cdots + n = O(n^2)$. So we conclude $1 + 2 + \cdots + n = \Theta(n^2)$.

Example 2. $1^k + 2^k + \cdots n^k = \Theta(n^{k+1})$ for any $k \geq 1$. Proof:

\[ 1^k + 2^k + \cdots n^k \geq (\lfloor \frac{n}{2} \rfloor)^k + \cdots + n^k \]
\[ \geq (\lfloor \frac{n}{2} \rfloor)^{k+1} \sim \frac{n^{k+1}}{2^{k+1}}. \]

Therefore, $1^k + 2^k + \cdots + n^k = \Omega(n^{k+1})$. On the other hand,

\[ 1^k + 2^k + \cdots + n^k < n^k + \cdots + n^k = n^{k+1}, \]

thus, $1^k + 2^k + \cdots + n^k = O(n^{k+1})$. We conclude $1^k + 2^k + \cdots + n^k = \Theta(n^{k+1})$.

We will do more examples in the forthcoming homework assignment.

6.2 Algorithms

Roughly, an algorithm is a step-by-step procedure involving a finite sequence of instructions to solve a given problem.

The typical structure is Input $\rightarrow$ Algorithm $\rightarrow$ Output.

The algorithm is precisely stated, and contains only finitely many steps. Intermediate results within the algorithm are uniquely defined, and depend only on inputs and results of the preceding steps.

An algorithm is naturally described in a way that makes it directly implementable as a computer program.
6.2.1 Pseudocode

Example 1. $a, b, c$ three numbers. Task: Find the largest one. Let $x$ denote a free variable which will ultimately be assigned the maximal value out of $a, b, c$.

\[
\begin{align*}
  x & := a \\
  & \text{if } x < b \text{ then} \\
  & x := b \\
  & \text{if } x < c \text{ then} \\
  & x := c 
\end{align*}
\]

Henceforth, we will use pseudocode for the description of algorithms because of precision, structure, and universality. Pseudocode resembles computer languages such as Pascal, Modula, C++.

The above example is then written as

\[
\text{procedure max}(a, b, c) \\
  x := a \\
  \text{if } b > x \text{ then} \\
  x := b \\
  \text{if } c > x \text{ then} \\
  x := c \\
  \text{return}(x) \\
\text{end max}
\]

The first line declares that the algorithm (referred to as procedure) is called max, and that the input of max is $a, b, c$. The symbol := is the assignment operator, it associates to the variable on its left hand side the value on its right hand side.

6.2.2 The if-then clause

The if-then clause has the form

\[
\text{if } p \text{ then} \\
  \text{action}
\]

That is, if $p$ is true, then action is carried out, and then the algorithm passes to the subsequent line. If, however, $p$ is false, action is left out, and the algorithm directly passes to the subsequent line.

Another if-then clause is given by

\[
\text{if } p \text{ then} \\
  \text{action}_1
\]
else

\textit{action}_2

In case \textit{action} consists of multiple commands, one 'brackets' the latter using 'begin' and 'end'. For example,

\begin{verbatim}
if p then
  begin
    \[ x := x + 2 \]
    \[ b := x - a \]
  end
\end{verbatim}

The 'return' command terminates a procedure, and returns the value of \( x \).

\subsection*{6.2.3 The while-loop}

The \textit{while}-loop has the structure

\begin{verbatim}
while \( p \) do
  \textit{action}
\end{verbatim}

That is, \textit{action} is repeated as long as \( p \) is true.

Example 2. Finde the largest element in a sequence \( s_1, \ldots, s_n \).
Input: Sequence \( s_1, \ldots, s_n \), length \( n \).
Output: Largest element \( x \) in this sequence.

\begin{verbatim}
procedure maxel(s_1, \ldots, s_n)
  \[ x := s_1 \]
  \[ i := 2 \]
  while \( i \leq n \) do
    begin
      if \( s_i > x \) then
        begin
          \[ x := s_i \]
          \[ i := i + 1 \]
        end
      end
    return(\( x \))
end maxel
\end{verbatim}

Please analyze this procedure step by step.
6.2.4 The for-loop

A for-loop has the following structure

\[
\text{for } i := i_0 \text{ to } n \text{ do action}
\]

Example 3. Same input, output as in example 2. But now, use for-loop, instead of if-then clause.

procedure maxel\((s_1, \ldots, s_n)\)
\[
x := s_1 \\
fors i = 1 \text{ to } n \text{ do} \\
\quad \text{if } s_i > x \text{ then} \\
\qquad x := s_i \\
\quad \text{return}(x)
\]
end maxel

We will in class see many more examples of algorithms. Please familiarize yourselves with the above pseudocode language.

7 Recursive Algorithms

A recursive algorithm is a procedure that invokes itself.

Example 1. Computing \(n!\).
Input: \(n\).
Output: \(n!\).

procedure factorial\(n\)
\[
\quad \text{if } n = 0 \text{ then return}(1) \\
\quad \text{return}(n \cdot \text{factorial}(n - 1))
\]
end factorial

Please analyze this algorithm carefully. Take for example \(n = 3\). Then, the command return(1) will be ignored, and the procedure returns 3-factorial(2). The value of factorial(2) must then be evaluated. Here again, the command return(1) is ignored, because now, \(n = 2 \neq 0\), etc. After one more recursion, we have \(n = 1\). The command return(1) is ignored, and the procedure directly passes to return(1-factorial(0)). But for factorial(0), the case \(n = 0\) is given, and the command return(1) is executed, terminating the recursion.

Note that it is necessary to implement a command that defines when the procedure should stop invoking itself. Here, it is is the line declaring what
to do if $n = 0$.

Example 2. The Euclidean algorithm. Let the command "swap($a, b$)" mean that $a$ is assigned the value of $b$, and $b$ is assigned the value of $a$ (their respective values are swapped).

procedure gcd($((a, b))$)
    if $a < b$ then
        swap($a, b$)
    if $b = 0$ then
        return($a$)
    r := $a$ (mod $b$)
    return(gcd($b, r$))
end gcd

Analyze this recursive algorithm carefully! How does the procedure terminate? Carry out the example gcd(7, 12) step by step.