Homework 1

CS 3343 – Fall 2006 | assigned August 24, 2006
Tom Bylander, Instructor | due September 1, 2006

Your solutions must be submitted as a document to WebCT. Part of what you should learn this semester is being able to write mathematical expressions using some word processing software. For example, both Microsoft Word and OpenOffice have equation editors.

I prefer to use Latex, which admittedly has a higher learning curve than Word or OpenOffice. Applying the command `pdflatex` to this file `hw1.tex` (which you can download from the web site), results in the PDF file `hw1.pdf`.

1. (20 pts.) Prove the equality $\gcd(m, n) = \gcd(n, m - n)$ for positive integers m and n, assuming $m \geq n$.

2. (20 pts.) Do Exercise 1.2.9, making at least one improvement.

   Consider the following algorithm for finding the distance between the two closest elements in an array of numbers.

   algorithm *MinDistance*($A[0..n-1]$)
   // Input: Array $A[0..n-1]$ of numbers
   // Output: Minimum distance between two of its elements
   $dmin \leftarrow \infty$
   for $i \leftarrow 0$ to $n-1$ **do**
   for $j \leftarrow 0$ to $n-1$ **do**
   if $i \neq j$ **and** $|A[i] - A[j]| < dmin$ **then**
   $dmin \leftarrow |A[i] - A[j]|$
   return $dmin$

   Make at least one improvement in this algorithm.

3. (20 pts.) Do Exercise 1.3.1, but, for part a, sort the list $47, 14, 98, 81, 35, 60$ instead and be sure to show the values in the *Count* array.

   Consider the following algorithm for the sorting problem that sorts an array by counting, for each of its elements, the number of smaller elements and then uses this information to put the element in its appropriate position in the sorted array:

```
algorithm ComparisonCountingSort(A[0..n − 1])
    // Sorts an array by comparison counting
    // Input: Array A[0..n − 1] of orderable values
    // Output: Array S[0..n − 1] of A's elements sorted in nondecreasing order
    for i ← 0 to n − 1 do
        Count[i] ← 0
    for i ← 0 to n − 2 do
        for j ← i + 1 to n − 1 do
            if A[i] < A[j] then
                Count[j] ← Count[j] + 1
            else Count[i] ← Count[i] + 1
    for i ← 0 to n − 1 do
        S[Count[i]] ← A[i] return S
```

(a) Apply this algorithm to the list $47, 14, 98, 81, 35, 60$.

(b) Is this algorithm stable?

(c) Is it in place?

4. (20 pts.) Do Exercise 1.4.4.

   (a) Let A be an adjacency matrix of an undirected graph. Explain what property of the matrix indicates that

      i. the graph is complete.

      ii. the graph has a loop, i.e., an edge connecting a vertex to itself.

      iii. the graph has an isolated vertex, i.e., a vertex with no edges incident to it.

   (b) Answer the same questions for the adjacency list representation.

5. (20 pts.) Do Exercise 1.4.6.

   Prove the inequalities that bound the height of a binary tree with n vertices:

$$\lfloor \log_2 n \rfloor \leq h \leq n - 1$$

   To show the lower bound, show that a binary tree of height h can have up to $2^{h+1} - 1$ vertices (use summation formula 5 on p. 470, which is also the 4th summation formula on slide 13 of my Chapter 2 notes). This implies that one more vertex requires a binary tree of height $h + 1$. Show how the lower bound corresponds to this property.