Computer Science 3731/6906 **Fall 2018**

Assignment 8

Date due: November 14 (in class) See Assignment 2 for instructions on electronic submission.

---

1. Write a Fortran recursive integer function *GCD (m,n)* which returns the greatest common divisor of positive integer numbers m and n. The recursive definition is as follows:

$$GCD(m,n) = \begin{cases} GCD(n,m), & \text{if } n > m; \\ m, & \text{if } n = 0; \\ GCD(n, \mathrm{mod}(m,n)), & \text{if } m > n \wedge n > 0. \end{cases}$$

   where the (Fortran) function $\mathrm{mod}(m,n)$ returns the remainder of dividing m by n.

   Write also a Fortran program which reads a sequence of integer numbers and uses *GCD(m,n)* to find tghe greatest common divisor of the numbers in the sequence.

2. Adjacency matrix of a finite directed graph with n vertices is a square matrix A of $n \times n$ elements, in which nondiagonal entry $a_{i,j}$ is 1 if there is an edge from node i to node j in the graph, otherwise $a_{i,j}$ is zero. A diagonal entry $a_{i,i}$ is 1 only if there is a loop on node i (i.e. an edge from i to i). Write a Fortran logical function $StronglyConn(A, n)$ which checks if a directed graph described by its adjacency matrix $A[1:n, 1:n]$ is strongly connected (a strongly connected graph contains a path connecting any pair of nodes).

   Also, write a Fortran program which enters a directed graph, for example as a sequence of edges (i.e., pairs of nodes) and creates the adjacency matrix, invokes $StronglyConn$ and outputs its result.

   For example, for a 4–node graph with 6 edges:

```
1,2
1,4
2,3
3,1
3,4
4,2
```

   the adjacency matrix is:

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

   This graph is strongly connected.

   Hint: To check if there is a path from vertex i to any other vertex of a graph, an auxiliary n-element array X can be used which initially contains a copy of row i from the adjacency matrix A, and to which iteratively are added rows of A indicated by nonzero elements of X. When no new elements are added to X, the iteration ends. All nonzero elements of X indicate those nodes to which there is a path from node i.

   To check strong connectivity, the procedure is repeated for all nodes i of the graph.