CSCI 340 **Computer Assignment 10** **Fall 2017**
(10 points)

Weighted Digraphs

For this computer assignment, you are to write a C++ program to implement several graph algorithms on weighted digraphs. These digraphs are represented by the wdigraph class, which is defined in the header file wdigraph.h, in directory: /home/cs340/progs/17f/p10.

Each digraph is created randomly by the constructor of the class. You can find the implementation of the constructor in the source file wdigraph1.cc, which is also in the same program directory.

For a given digraph size of size, each node of a digraph can be referenced by index values 0, 1, ..., (size −1); however, nodes of a digraph are labeled consecutively by capital letters, starting from A, which corresponds to the index value 0. Use index values when you reference a node in a digraph but node labels in your printing.

Implement the following member functions of the wgraph class and put your implementations of these functions in your source file wdigraph2.cc. Insert the line #include "/home/cs340/progs/17f/p10/wdigraph.h" at the top of this source file.

- void wdigraph :: depth_first (int u) const: It traverses a digraph in the *depth-first order* and it also prints the resulting path. The index value u of the starting node is given as an input argument to this function.

- void wdigraph :: print_graph ( ) const: It prints the followings for a digraph:

  1. Number of nodes in the digraph.
  2. An adjacency matrix for the digraph.

  In printing, if nodes i and j are adjacent to each other, insert the weight factor of the link (i, j) in row i and column j of the adjacency matrix; otherwise, simply insert the '–' character in that position.

The source file of the driver program prog10.cc is in the same program directory with wdigraph.h. The main ( ) routine includes the subroutine: void proc_graph (wdigraph& g) to test the digraph g as follows:

1. Prints the digraph by calling the member function print_graph ( ).
2. Traverses the digraph using the depth-first search and prints the resulting paths by calling the member function depth_first ( ); however, to save space, it only prints the paths for every M-th node for M = 3. In other words, it only prints the nodes, starting with the index values 0, 3, 6, ...

The main ( ) routine creates a digraph for the default size of NO_NODES (defined in the header file wdigraph.h) and then creates two other digraphs of sizes N2 = 5 and N3 = 20.

For all cases, it calls the subroutine proc_graph ( ) to test the member functions of wgraph.

To compile the driver program of the source file prog10.cc (contains the main ( ) routine and the subroutine proc_graph ( )), the source file wdigraph1.cc (contains the implementation of the class constructor and destructor) and the source file wdigraph2.cc (contains your implementation of member functions of wdigraph), and link the generated object files with the system library routines, execute: Make N=10. To test your program, execute: Make execute N=10. This will execute the program and generate the output on both your terminal screen and in output file prog10.out. After you are done with your program, you don't need its object and executable files any more. To delete them, execute: Make clean.

The correct output can be obtained from the file prog10.out, which is also in the same program directory with wdigraph.h.

When your program is ready, submit its source file to your TA, by executing: mail_prog.340 wdigraph2.cc.