This lab concerns directed acyclic graphs.

1. Get the program `dacgraph.c` and compile and link it. Get the sample datafiles `dacgraph.dat` and `dacgraph2.dat`. Read the introduction in the code `dacgraph.c` which explains the format of the datafiles. `metric` and `capacity` should be set to `1` for active edges. Future modifications of the program may make use of these values but they will not be used in the following exercise. Run the program on the sample datafiles first, for example:

   ```
   dacgraph dacgraph.dat
   dacgraph dacgraph2.dat
   ```

   You should ask the program to print the matrix (’m’) representing the directed graph and do a depth-first traversal (’t’).

2. Make your own datafiles for directed graphs and try the program on your datafiles. You should use 8–12 nodes; more than 12 will not display well on 80-columns. Try to fool the program by giving it a directed graph which is not acyclic. What happens? Look at the code, particularly the routine `depth_first_traversal()` and the stack operations `init_stack()`, `push_stack()`, etc..

**Assignment** The assignment is to turn in your answers to each of the following three problems. Email me your results as *in-line text*. Please:

   i. keep line length under 72 characters per line.

   ii. do not send me any attachments.

1. Where is the code which enables the program `dacgraph.c` to find out which nodes are “startnodes” for depth-first traversal? How does it work?

2. Given a directed acyclic graph with $n$ nodes, how can we tell what the maximum depth of the stack could be during a depth-first traversal and why?

3. What is the field `active_onstack` in the `struct nodeinfo` used for? Be specific about where and how it is used and why it is necessary.