Midterm 1

Name:

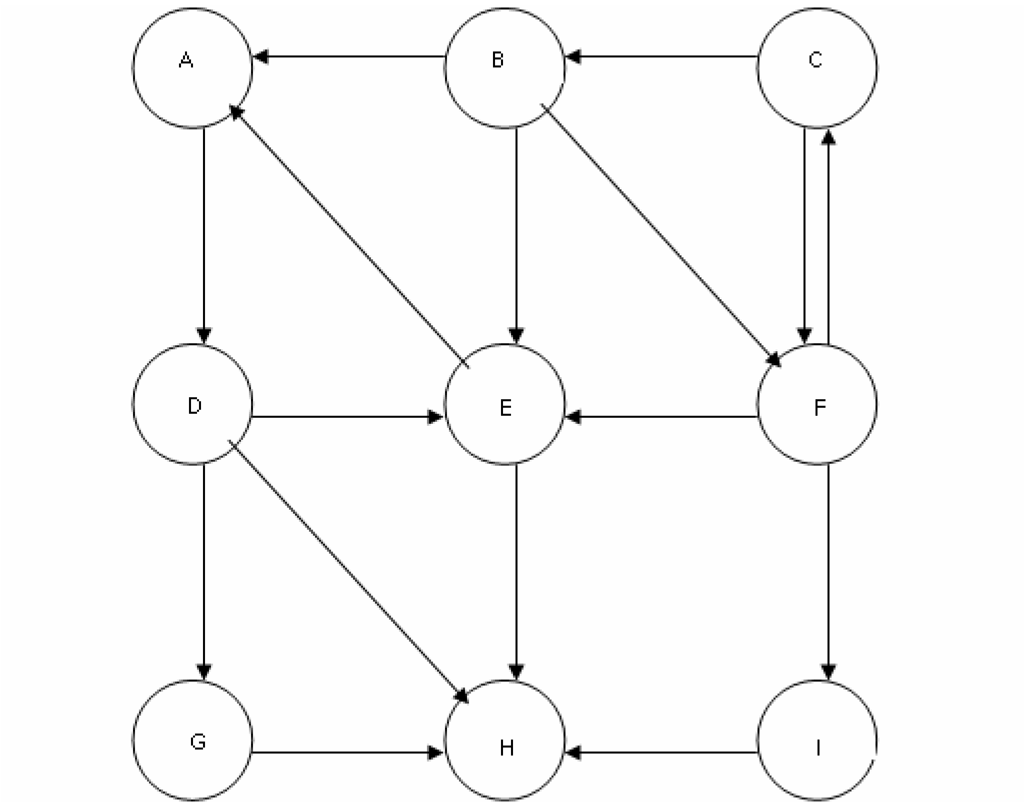
TA:

Answer all questions. Read them carefully first. Be precise and concise. The number of points indicate the amount of time (in minutes) each problem is worth spending. Write in the space provided, and use the back of the page for scratch. Good Luck!

1	
2.1	
2.2	
2.3	
2.4	
2.5	
2.6	
2.7	
2.8	
2	
3	
total	

Problem 1

(15 points) Do a depth-first search of the graph below. Follow our convention of processing nodes in lexicographic order. Show the pre and post order numbers. How many strongly connected components are there (no need to show how to find them). What is the graph after you shrink them?



Problem 2

In each of the following questions, provide a brief explanation. The space provided should suffice.

1. (5 points) What happens when Euclid's algorithm is called with arguments F_n, F_{n+1}, two consecutive Fibonacci numbers?

2. (5 points) How many integers modulo $1331 = 11^3$ have inverses?

3. (5points) In an RSA cryptosystem, p = 7 and q = 11. Find appropriate exponents d and e.

4. (5 points) What is $2^{2^{46}}$ (mod 47)?

5. (5 points) How many lines, as a function of n (a power of 2) in Θ form, will this program print? Write a recurrence and solve it.

```
boring(n)
if n > 1 do:
{print("are we done yet?");
boring(n/2);
boring(n/2)}
```

6. (5 points) We denote the discrete Fourier transform of (1, 0, 1, -1) by (A_0, A_1, A_2, A_3). What is ω in this case? And what is A_3?

7. (5 points) The FFT and its inverse are based on certain properties of (equations satisfied by) ω. What are they?

8. (10 points)

   Is each of the following statements true or false? Explain briefly.

   (1) If the depth-first search of a directed graph has a cross edge, the graph is not strongly connected.

   (2) If a graph is not strongly connected, then its depth-first search must have a cross edge.

Problem 3

(20 points) You are given a sorted array of distinct integers a[0],…,a[n-1], positive and negative, and you want to find out whether there is an index I such that a[i] = i. Give a divide-and-conquer algorithm for this problem. What is the running time of your algorithm? (It should be as little as possible.) Make sure you prove the correctness of your algorithm and of your claim to running time.