CS 373: Combinatorial Algorithms, Summer 2000
IMCS/Quantum

Midterm 2 (Wednesday July 19, 2000)

Name:
Net ID:

This is a closed-book, no-calculator exam!

You may use an $8\frac{1}{2}'' \times 11''$ sheet of notes (both sides). **Please turn this in with your exam**.

- You have 75 minutes to complete the exam.
- Print your name, netid, and alias in the boxes above, and print your **netid** at the top of every page.
- Answer all the questions. They are ten (10) points each.
- For questions that ask for you to design and analyse an algorithm, you need to describe the algorithm in English, justify its correctness, and analyse its running time to get full credit. Also, to get full credit, give an algorithm that is as efficient as possible.
- Make sure you justify your answers: a correct algorithm with no justification of its correctness, will only get partial credit; just pseudocode may get no points at all.
- If you need more than the front of the page, please mark clearly that the answer is continued on the back of that page. If you feel like you want to erase something, don't. Just cross it out and plainly mark your official answer.

#	Score
1	
2	
3	
4	
5	
Total	

1. **Multiple Choice**

   Every question below has one of the following answers.

   (a) $O(1)$ (b) $O(\log V)$ (c) $O(V)$ (d) $O(V \log V)$ (e) $O(V^2)$ (f) $\omega(V^2)$

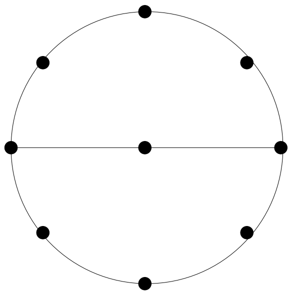
   For each question, write in the box the letter that corresponds to the **best** answer. You do not need to justify your answer. Each correct answer earns you 1 point; incorrect answers will not penalize you.

   Note that V and E refer to the number of nodes and edges respectively in an arbitrary *undirected* graph with adjacency list representation, or in the case of a set of points or values, V refers to the number of points or values. $g(n) = \omega(f(n))$ means that $g(n)$ is **strictly** greater, asymptotically, than $f(n)$.

   - [ ] $O(E - V)$.
   - [ ] $O(\log E)$.
   - [ ] Computing shortest paths on a graph with no cycles.
   - [ ] Determining if a graph has a cycle of length 3.
   - [ ] Determining if a graph is disconnected.
   - [ ] Determining if a graph has more than 10 nodes.
   - [ ] Determining if a graph has less than 10 edges.
   - [ ] Determining for three points A, B, C, whether they are in counterclockwise order.
   - [ ] Out of V points, determining the four points having minimum and maximum x and y coordinates.
   - [ ] If the load factor α is $\log V$, the worst case time for a search in a closed hashing table with V elements.

2. **Counting Spanning Trees**

A "steering-wheel" graph of order n, called SW_n, is a cycle of length $2n$ plus an additional node (called the hub, h) with an edge to 2 nodes in the cycle that are furthest apart. The following is a picture of SW_4:



(a) (5 pts) Prove that the number of spanning trees on SW_n is $n^2 + 4n$.

(b) (5 pts) Design and analyse an algorithm to compute the minimum spanning tree of SW_n (you may assume that you know which node is the hub, h).

3. **Dynamic update of shortest paths**

   Suppose you have a graph G and a node s in that graph for which you have already computed shortest paths. Then you are told that one of the weights in the graph has changed. Design and analyse an algorithm that updates the shortest paths tree.

4. **Convexity**

   Design and analyze an algorithm to determine whether a simple n-vertex polygon is convex.

5. **Diameter of a set of points**

   Design and analyze an algorithm that finds a pair of vertices of an n-vertex convex polygon that are furthest apart.