COS 528: ***Data Structures & Graph Algorithms*** **Homework**
Fall, 1997

Your objective will be to earn 100 points for the semester. I will provide a variety of ways to do this, including solving problems, doing research, and writing survey/tutorial papers. Each exercise will have a due date. Please do your graded work entirely on your own.

1. Research/Tutorial Paper: worth up to 80 points. Due by the end of the semester.

   Write a paper exploring the meaning of "functional programming" and its variants in the context of efficient data structure and algorithm design. Some questions to ponder:

   What is functional programming? Strictly functional programming? What is memoization? What is the power of memoization? What is the power of write-once memory vs. strictly functional programming vs. arbitrarily writeable memory?

2. Exercise: Single-pair bottleneck shortest paths: worth up to 20 points. Due by 11/6.

   Given a directed graph with real-valued edge capacities, and two distinguished vertices, s and t, find a path from s to t to whose minimum edge capacity is as large as possible. Describe a linear-time algorithm for this problem.

3. Exercise: Solve the bottleneck path problem for a single source s and *all* possible sinks t in linear time (or as fast as you can), using only comparisions as operations on the edge capacities. Worth up to 60 points. Due by the end of the semester.

4. Exercise: Shortest paths with edge-length bounds. Due by 11/6. Worth up to 20 points. Consider the single-source shortest path problem with non-negative edge lengths, such that each edge length is either zero or in the interval $[L, U]$, where $0 < L < U$. Describe how to solve the shortest path problem in $O\ (M + n(U/L))$ time.

5. Research problem: Improve the bound in problem 4 to $O(m + n \log(U/L))$ or better (assume $U \geq 2L$). Worth up to 40 points. Due by the end of the semester.

6. This and #7 refer to the paper "Simple Confluently Persistent Catenable Lists," by Kaplan, Okasaki, and Tarjan.

   Exercise: Consider the "double-sublist" representation of catenable deques described in Section 5. Show how to simplify this structure so that first middle buffers and last middle buffers are each of size 2 or 3, prefixes and suffixes are each of size 3-6, and the amortized time per deque operation is still constant. Worth up to 20 points. Due by 11/20.

7. Research problem: Show that the "single-sublist" representation of catenable deques, sketched in Section 6, or a variant thereof, has an amortized constant time bound per deque operation. Worth up to 80 points. Due by the end of the semester.