Randomized Algorithms

close-book midterm exam

April 28, 2004

Instruction Please write down your name and ID on each piece of your answer sheets. Cheating will be most seriously punished. Any dishonest attempt in this exam implies an 'F' as your final grade.

You may use anything of our lectures as a subroutine to your answers, unless you are explicitly asked to explain something we explained in class.

Problem 1 (20 points)

Explain what are

- (5 points) randomized algorithms,
- (5 points) probabilistic analysis of algorithms,
- (5 points) probabilistic methods, and
- (5 points) non-uniform algorithms.

Problem 2 (20 points)

- (5 points) State Yao's Minimax Principle (i.e., Yao's Inequality).
- (15 points) Use Yao's Minimax Principle to prove that the (worst-case) expected running time of any Las Vegas algorithm for sorting n numbers by comparison is $\Omega(n \log n)$. You may assume $\log_2 n! = \Omega(n \log n)$.

  Hint: you might want to *prove* and use the fact that the average depth of leaves in a binary tree with $n!$ leaves is $\Omega(n \log n)$.

Problem 3 (15 points)

Recall that Valiant and Brebner's permutation routing algorithm in an n-cube has two phases. In the first phase, the packet on each node is routed to a randomly chosen intermediate node in the hypercube via the *bit-fixing path*. You are asked to prove that for each edge e in the n-cube, the expected number of these 2^n bit-fixing paths that pass edge e is exactly 0.5.

Problem 4 (15 points)

We showed in class a randomized approximation algorithm for MAXCUT. Recall that the algorithm outputs each *node* independently with probability 0.5. As we have seen, it is straightforward to prove that the expected approximation ratio of this algorithm is at least 0.5.

Now, you are asked to derandomize this randomized algorithm into a deterministic polynomial-time algorithm whose approximation ratio is at least 0.5. (Don't just give your derandomized algorithm. You have to explain why the algorithm is indeed obtained from derandomizing the above randomized approximation algorithm.)

Problem 5 (10 points)

Do you think the following situation is possible? Justify your answer.

> Π is an NP-complete minimization problem. (For example, Vertex Cover is a possible candidate for Π. Minimum Cut is not, since it is not NP-complete. Neither are MAXCUT and MAXSAT, since they are maximization problems.) Algorithms Dumb and Dumber are two polynomial-time approximation algorithms for Problem Π. The approximation ratio of Algorithm Dumb is $\rho > 1$ and the ratio is tight. The approximation ratio of Algorithm Dumber is $\rho' > 1$ and the ratio is tight. Now, Algorithm Clever is a randomized algorithm which runs Algorithm Dumb or Dumber randomly and equally likely. Do you think it is possible that the expected approximation ratio is strictly less than $\min(\rho, \rho')$?

Problem 6 (10 points)

Consider random walks on an n-node cycle C_n. Let $v_1, v_2, \ldots, v_n$ be the nodes on C_n around the cycle. What is the hitting time $H(v_1, v_i)$ on C_n from v_1 to v_i for each $i = 1, 2, \ldots, n$? (Don't just give your answer. Show how you obtain it.)

Problem 7 (10 points)

- (5 points) State Algorithm RandQS (i.e., randomized quick-sort) we see in class.
- (5 points) Show that the expected running time of Algorithm RandQS on n numbers is $O(n \log n)$.