Theory Qual: Date: Monday September 15, 2003

Please answer all questions. TWO pages.

1. This is a problem in computer disk caching. There are n distinct pages, P_i where $i = 1, 2, \ldots, n$. Each page P_i has an integer size $s_i > 0$, and an integer cost $c_i > 0$. There is a cache which has total size S. Initially, the cache is empty.

   A sequence of page accesses $i_1, i_2, \ldots, i_m$ is given, which must be processed in order. To process a page P_{i_j}, if the page is not already in the cache, it needs to be fetched from disk at a cost of c_{i_j}. If it is already in the cache, then it is free. After processing a page, one has the option of evicting any number of pages from the cache, and possibly retaining the newly fetched page in the cache. These operations incur no cost. At any point in time, the total size of the pages in the cache must be no more than S.

   Problem: Given the data $s_i, c_i, S, i_1, i_2, \ldots, i_m$, (all integers are written in binary) find a sequence of paging operations which minimizes the total fetching cost.

   Prove this problem is NP-hard.

   (Hint: Find a reduction from the Knapsack Problem.)

2. We are given a list of n different numbers, $X_1, X_2, X_3, \ldots X_n$. Patience sorting is the following process, which works with up to n stacks.

   For $i = 1$ to n

   Find the first stack whose top is greater than X_i, and add X_i to the top of that stack.

   If no such stack exists, start a new stack with X_i on top, placing it to the right of the existing stacks.

   For example, if we have the sequence 3 1 4 2 5 9 the stacks are

	1	1		1	2	1	2		1	2		
3	3	3	4	3	4	3	4	5	3	4	5	9

   a) Prove that if $Y_1 < Y_2 < \ldots < Y_k$ is an increasing subsequence of $X_1, \ldots, X_n$, then the Y_i's are placed in distinct stacks.

   b) Show that there is an increasing subsequence of the X_i's whose length equals the final number of stacks. (Hint: when placing a number on some stack other than the first, provide a pointer to the number currently on top of the stack immediately to its left.)

   c) Show that a longest increasing subsequence of $X_1, \ldots, X_n$ can be found using $O(n \log n)$ steps.

3. Suppose there exists a randomized polynomial-time algorithm with the following behavior: On input an unsatisfiable Boolean formula, it outputs "no" with probability at least 3/4; on input a Boolean formula with exactly one satisfying assignment, it outputs "yes" with probability at least 3/4; on any other input, the behavior of the algorithm is unspecified. Prove that the existence of such an algorithm implies that the polynomial-time hierarchy collapses. To which level can you prove the PH collapses to?

4. A planar graph is a graph that can be embedded in the plane such that no two edges cross. The planar separator theorem gives an algorithm that, on input a planar graph on n vertices, outputs in time $O(n)$ a partition of the vertices into three sets, A, B, and C such that A and B are of size at most $2n/3$, C is of size at most $4\sqrt{n}$, and there are no edges between A and B.

   Construct an algorithm that, given a planar graph on n vertices, finds the size of a largest independent set in time $2^{O(\sqrt{n})}$.

   What if you consider cliques instead of independent sets?

Please be concise.