CS 182
Advanced Topics in Algorithms
Spring 2003
Problem Set 3a
Due Thursday, February 6 in class

1. [**15 Points**] **Union-Find with Partial Path Compression.** In class we examined the union-find data structure using union-by-size and path compression. One disadvantage of path compression is that it is a two-phase process: First we must "climb" the path to find the root. Then, we go back and set the parent of each node to be the root.

   A simpler approach is to use *partial path compression* in which each node on the path has its parent pointer changed to point to its grandparent. This process can be done in just one pass as we climb up the path. Show that using partial path compression instead of full path compression still allows us to perform any sequence of n MAKESET, UNION, and FIND operations in time $O(n \log^* n)$ time.

2. [**15 Points**] **Union-Find Again.** Consider again the union-find data structure using union-by-size and path compression. Assume that we will perform a sequence of n MAKESET, UNION, and FIND operations where all of the MAKESETs are performed before the UNIONs and all of the UNIONs are done before the FINDs. Show that the total running time is now asymptotically even better than $\Theta(n \log^* n)$.