Algorithms	**Problem Set 3**
CS 6820 Fall 2009	**Due October 9, 2009**

Recall the general instructions for handing in homework:

- If possible, please typeset the homework (i.e. format your solutions as an electronic file using latex or Word with mathematical notation).
- Homework solutions done electronically can be handed in by directly uploading them to CMS. Please mail Ashwin (`ashwin85@cs.cornell.edu`) if you have any trouble with this.

This problem set consists of two problems that you should prove NP-complete. For this purpose, you can use a reduction that involves any of the problems discussed in lecture or in Chapter 8 of the book.

(1) Consider the following problem related to the analysis of datasets encoding on-line behavior. Suppose that a large search company wants to release an anonymized dataset consisting of queries issued by registered users — those who were issuing the queries while signed into their accounts with the company. Each line of the file will consist of an (anonymized) user ID followed by a query. Thus, for example, the set of lines

```
u0000 movie downloads
u0000 song lyrics
u0000 twitter apps
u0001 np-completeness
u0001 3-sat
u0002 arsenal
u0002 man united
```

specifies three queries by the user with ID `u0000`, two queries by the user with ID `u0001`, and two queries by the user with ID `u0002`. Of course, the real file is enormously longer than this.

Now, for a given user u, we say that a set of queries $Q = \{q_1, q_2, \ldots, q_k\}$ is a *distinguishing set* for u if user u has issued each of the queries in Q (as recorded by the data), and there is no other user u' who has also issued each query in Q. That is, in order for these queries to be a distinguishing set for u, the data should contain the lines

u q_i

for each i, and there should be no other user u' with this property.

If you're in the data, and there's a small distinguishing set for you, then this could be cause for worry: for example, if someone happens to know that you've searched for both "np-completeness" and "3-sat" on the search engine releasing the data, and there's only one user in the data who issued both these queries, then there's reason to suspect that you're this user, in which case all your other queries are revealed by the data as well. (Of course, you still might not definitely be this user — it depends on whether the dataset being released includes *all* queries from a given time period, and whether your queries in the distinguishing set came from this time period. But in this question we're focusing only the formal definition of a distinguishing set, which is self-contained, rather than these possible implications of the definition.)

Show that the following problem is NP-complete: given a dataset as above, a user u, and a number k, does there exist a distinguishing set for user u of size at most k?

(2) Here's a problem that David Liben-Nowell and I encountered while reconstructing the trees by which chain-letter petitions spread on the Internet. Suppose you are given many copies of a petition that circulated on-line. Each copy has a list of names of people who signed that copy, in order. So for example, we might find four copies whose lists (numbered in order) look as follows:

```
1. A, 2. B, 3. C, 4. D
2. A, 2. B, 3. C, 4. E, 5. F
3. A, 2. X, 3. C, 4. E, 5. F, 6. G
4. A, 2. B, 3. F, 4. H
```

Suppose we happen to know that the first two lists are "genuine" — in the sense that all these people actually signed in this order — while lists three and four have been "corrupted." In this case, the first two lists are easy to interpret: each begins with signatures by people A, B, and C in order; we would conjecture that C then sent the petition to two people, D and E. D signed one copy of the petititon, while E signed the other and passed it on to F.

The corrupted third and fourth copies show the kinds of mutations you tend get in on-line petitions. Lists like the third one in our example generally arise because someone (e.g. the person G) not only signed it, but also changed one of the names higher up in the list (in this case changing B to X). Lists like the fourth one in our example generally arise because someone (e.g. the person H) not only signed it, but also deleted a block of names from higher up in the list (in this case deleting the names C and E).

Now, of course, in reality we aren't told that certain lists are genuine and others are corrupted. Rather, we have a large collection of lists, some of which are corrupted, and we'd like to find a large subset that we can hypothesize are the genuine ones — in the sense that they're all consistent with a single uncorrupted pattern of spread for the chain letter.

Here's a way to make this precise. Suppose that we're given a collection of lists of names. We say that two lists are *consistent* with each other if they agree on some common prefix, and then contain disjoint sets of names after this point. So in our example above, lists 1 and 2 are consistent with each other. Note also that lists 1 and 4 are consistent with each other, since they agree on the first two names and then split after that. But lists 2 and 4 are not consistent with each other, so we can't for example conclude that the collection of lists 1, 2, and 4 is a set that's pairwise consistent — with consistency between each pair of lists in the collection.

Unfortunately, finding large sets that are pairwise consistent is hard. Specifically, show that the following problem is NP-complete: given a collection of lists of names, and a number k, decide whether there exists a subset S of k lists from the collection with the property that each pair of lists from S are consistent with each other.