Problem Set 1 **Due: Feb 3** **cs2me3 Winter 2004**

Instructions

1. You are encouraged to work in groups of three. If you cannot find two partners, you can work in a group of two. As a last resort, you can work alone.
2. The names of all the partners must be indicated on the assignment.
3. For **Part A** of the assignment, you must submit an electronic copy of your C program via WebCT, and a hard copy of your program should be included when you hand in the assignment.

Part A

1. Write a C program which for a given graph with (positive integer) weights on the edges outputs a minimum cost spanning tree.

   The input file must be of the following form:

$$\begin{array}{lll} N & & \\ i_1 & j_1 & w_1 \\ i_2 & j_2 & w_2 \\ \vdots & \vdots & \vdots \\ i_k & j_k & w_k \\ -1 & & \end{array}$$

   Here N is the number of nodes in the graph, and each line after the first line contains three positive integers (separated by a single space), where the first two represent an edge, and the last is the weight of that edge.

   Your program must take the file name of the file containing the data as input, and output the edges that make up a minimum cost spanning tree in the following format:

$$\begin{array}{ll} i_1 & j_1 \\ i_2 & j_2 \\ \vdots & \vdots \\ i_k & j_k \end{array}$$

   If no minimum cost spanning tree exists (if, for example, the graph is not connected), the program must output a message informing the user of this fact. The program must also check for the validity of the input (for example, that no i_p and j_p is bigger than N, and that there are no negative integers, and all the other bad cases that you can think of).

   For full marks, the program must compile, run on a set of test files, be well documented with comments, and well written (i.e., simple, clear, concise).

Part B

1. Show that every tree with n nodes has exactly $n - 1$ edges.
   (**Hint.** Induction on n, using the fact that every finite tree has a *leaf*, i.e., a node of degree one.)

2. Show that a graph with n nodes and more than $n - 1$ edges must contain at least one cycle.

3. Prove the **Exchange Lemma:** Let G be a connected graph, and let T_1 and T_2 be any two spanning trees for G. For every edge e in $T_2 - T_1$, there is an edge e' in $T_1 - T_2$ such that $(T_1 \cup \{e\}) - \{e'\}$ is a spanning tree for G.
   (**Hint.** Let e be an edge in $T_1 - T_2$. Then $T_2 \cup \{e\}$ contains a cycle—can all the edges in this cycle belong to T_1?)

4. Recall the quick-union (slow-find) program presented in class:

```
#include <stdio.h>
#define N 10000
main()
{
    int i, j, p, q, t, id[N];
    for (i=0; i<N; i++)
      id[i]=i;
    while (scanf("%d %d\n", &p, &q)==2) {
      for (i=p; i!=id[i]; i=id[i]);
      for (j=q; j!=id[j]; j=id[j]);
      if (i==j)
        continue;
      id[i]=j;
      printf(" %d %d\n", p, q);
      }
}
```

   Modify this program, by adding an array `sz[i]` which for each object with `id[i]==i`, maintains the number of nodes in the associated tree. Then, `sz` is used in the union operation so that the smaller tree is linked to the larger, thus preventing the growth of long paths in the tree.