Programming Comprehensive Exam (1996-1997)

1 Problem Description

The purpose of the program is to solve the warehouse location problem. The problem consists of selecting a number of warehouses to deliver goods to a number of customers in order to minimize the fixed costs and the transportation costs. Each warehouse has a fixed cost (for building it and maintaining it) and there are also transportation costs to deliver the goods from warehouses to customers. Obviously, when many warehouses are selected, the transportation costs decrease and the fixed costs are high. When few warehouses are selected, the fixed costs are low but the transportation costs increase. This tradeoff makes the problem NP-Complete.

The input data for a warehouse location problem consists of the following data items:

1. a number of warehouse locations, each of which associated with a fixed cost.

2. a number of customers: for each customer, the transportation from each warehouse is given.

The output data consists of the following results:

1. The list of warehouses which are selected;

2. The warehouse associated with each customer.

Note that, since we impose no capacity on the warehouses, the selection of the warehouse for a customer is easy once the warehouses have been selected.

Your program should implement a branch and bound algorithm for solving the warehouse location problem. In the branch and bound, a node represents a configuration $\langle S, R, U\rangle$, where S represents the set of selected warehouses, R represents the set of rejected warehouses, and U represents the set of warehouses which have not yet been considered. The branch operation of the algorithm selects a node $\langle S, R, U\rangle$, chooses a warehouse w in U, and generates two subproblems $\langle S\cup\{w\}, R, U\setminus\{w\}\rangle$ and $\langle S, R\cup\{w\}, U\setminus\{w\}\rangle$. It is your responsibility to find a bounding procedure adequate to avoid exploring the whole space of configurations. The use of a depth-first strategy for exploring the search space is recommended.

2 Input description

The program must obey the input conventions described in this section. Programs violating them will not be graded and will be awarded an NC. The object code should be called *locator* and must receive a file name as first parameter. The first two lines of the input file contain the number of warehouses (say n in the following) and the number of customers (say m in the following). The next two n lines contains the description of warehouses 1 to n. For each of them, we first give a name and then a fixed cost. The rest of the file is a series of m blocks, each block being associated with a customer. A block consists of n lines describing the transportation cost to warehouses 1 to n. All the numbers are assumed to be long integers.

3 Example of Input

Your program will be tested on data which are variations of the files

```
~pvh/COMPS97/data
~pvh/COMPS97/data1
```

No results for these files will be given. It is your responsibility to test your program.

4 Efficiency

Your program should be reasonably efficient. For the first benchmark, it should take no more than 8 seconds on a SPARC-10 and it should take no more than 15 seconds on the second benchmark. In general, it should take no more than 20% slower than my slow prototype. There is no guarantee on the grade for programs failing these efficiency criteria.

5 Documentation

Your program should be documented in such a way that it could be maintained by someone else. The files should be organized in a meaningful way and each function or method should be documented. You should also produce a document explaining the basic algorithm used in your program and the overall structure of your program (e.g., the purpose of each file).

6 Programming Languages

Use only C, C++, or Pascal to implement your program. The use of C++ is strongly recommended.

7 Handing the Program

On Friday 1/17/97, between 12pm and 6pm, you should copy your program and its documentation in the directory pvh/COMP97/Uxxx, where xxx is your number. Set the permissions such that the directory can be read and executed by everybody. Once again, failure to do so would give you automatically an NC. Note that if you want your program to be graded anonymously, you should have someone else copy it in the directory; otherwise, I will be able to know who you are.

8 Availability

I will be available Monday to answer questions and then every day of the week between 4pm and 5pm.