CISC621 Algorithm Design and Analysis, Fall 2015
Homework set 4, due Wednesday, November 11, 5:30pm

Check the "homework sheet" from the syllabus for general homework details. In particular, each homework solution is on an entirely separate (set of) sheet(s) of paper and is identified with your name(s). Do not staple solutions to two or more problems together. Submit in 201 Smith Hall directly to Chunbo Song or place on shelf marked CISC 621.

10. [Individual Problem] The Cantina of Babel problem (problem C on the next page) was on was on a recent programming contest. Let us generalize the problem in that we don't limit the number, N, of characters. Rather, we will bound the runtime of our solution as a function of N. In addition we won't put a bound on the number of languages a character understands. Finally we will add a feature to the situation. Let us say that conversing group A intimidates conversing group B if some member of A can be understood by some member of B (one way communication). The idea is that tension between members of these groups can be kept under control because the threat of retaliation by the entire group A can be communicated to any member of group B inhibiting their inclination to cause trouble. Solve the problem when the group allowed to stay in the bar consists of a conversing group A along with any *smaller* groups it can control by intimidation (direct or indirect – if indirect, each link must be intimidating (from larger to smaller)). Give detailed pseudocode, explain it, analyze your solution.

11. [Group Problem] The circuit counting problem (problem D, last page of this document) was on a recent programming contest. Let us generalize the problem slightly. We won't limit the number of vectors N by $N \leq 40$. Rather, we will analyze the runtime of our solution as a function of N. Also we won't restrict our vector coordinates to $|x|, |y| \leq 10$. Rather, we will specify $|x|, |y| \leq d$, where d is another problem parameter. Design an efficient algorithm to solve the circuit counting problem and analyze it's complexity. [Remark: My solution is in O(n^2d^2). Possibly you can do better.]

12. [Group Problem] The family of scorpion graphs is a well known example in the study of costs associated with the adjacency matrix representation of graphs. See, for example, http://www.cs.cornell.edu/courses/cs681/2007fa/Handouts/scorpion.pdf for discussion of the scorpion graph problem.

    Our scorpions have been subjected to radiation causing them to be born with multiple tails and stingers. Here's the definition: A k-scorpion is a graph that has k sting-tail pairs such that each sting is connected only to it's associated tail, which has degree 2. That tail's other connection is to the body. The body has degree n-k-1, being connected to every node except itself and the k stings. The remaining nodes (legs) are connected arbitrarily. Give an algorithm, running in O(kn) time, to determine if a graph given by adjacency matrix is a k-scorpion or not.

Problem C
Cantina of Babel

Photo by Brickset

Characters in Star Wars each speak a language, but they typically understand a lot more languages that they don't or can't speak. For example, Han Solo might speak in Galactic Basic and Chewbacca might respond in Shyriiwook; since they each understand the language spoken by the other, they can communicate just fine like this.

We'll say two characters can *converse* if they can exchange messages in both directions. Even if they didn't understand each other's languages, two characters can still converse as long as there is a sequence of characters who could translate for them through a sequence of intermediate languages. For example, Jabba the Hutt and R2D2 might be able to converse with some help. Maybe when Jabba spoke in Huttese, Boba Fett could translate to Basic, which R2D2 understands. When R2D2 replies in Binary, maybe Luke could translate to Basic and then Bib Fortuna could translate back to Huttese for Jabba.

In Star Wars Episode IV, there's a scene with a lot of different characters in a cantina, all speaking different languages. Some pairs of characters may not be able to converse (even if others in the cantina are willing to serve as translators). This can lead to all kinds of problems, fights, questions over who shot first, etc. You're going to help by asking some of the patrons to leave. The cantina is a business, so you'd like to ask as few as possible to leave. You need to determine the size of the smallest set of characters S such that if all the characters in S leave, all pairs of remaining characters can converse.

For example, in the first sample input below, Chewbacca and Grakchawwaa can converse, but nobody else understands Shyriiwook, so they can't converse with others in the bar. If they leave, everyone else can converse. In the second sample input, Fran and Ian can converse, as can Polly and Spencer, but no other pairs of characters can converse, so either everyone but Polly and Spencer must leave or everyone but Fran and Ian.

Input

Input starts with a positive integer, $1 \le N \le 100$, the number of characters in the cantina. This is followed by N lines, each line describing a character. Each of these N lines starts with the character's name, then the language that character speaks, then a list of 0 to 20 additional languages the character understands but doesn't speak. All characters understand the language they speak. All character and language names are sequences of 1 to 15 letters (a-z and A-Z), numbers, and hyphens. Character names and languages are separated by single spaces.

Output

Print a line of output giving the size of the smallest set of characters S that should be asked to leave so that all remaining pairs of characters can converse.

Sample Input 1

```
7
Jabba-the-Hutt Huttese
Bib-Fortuna Huttese Basic
Boba-Fett Basic Huttese
Chewbacca Shyriiwook Basic
Luke Basic Jawaese Binary
Grakchawwaa Shyriiwook Basic Jawaese
R2D2 Binary Basic
```

Sample Output 1

```
2
```

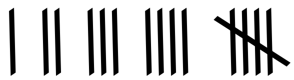
Sample Input 2

```
6
Fran French Italian
Enid English German
George German Italian
Ian Italian French Spanish
Spencer Spanish Portugese
Polly Portugese Spanish
```

Sample Output 2

```
4
```

Problem D
Circuit Counting



Suppose you are given a sequence of N integer-valued vectors in the plane (x_i, y_i), $i = 1, \ldots, N$. Beginning at the origin, we can generate a path by regarding each vector as a displacement from the previous location. For instance, the vectors $(1, 2)$, $(2, 3)$, $(-3, -5)$ form the path $(0, 0), (1, 2), (3, 5), (0, 0)$. We define a path that ends at the origin as a *circuit*. The example just given is a circuit. We could form a path using any nonempty subset of the N vectors, while the result (circuit or not) doesn't depend on the ordering of the subset. How many nonempty subsets of the vectors form circuits?

For instance, consider the vectors $\{(1, 2), (-1, -2), (1, 1), (-2, -3), (-1, -1)\}$ From these vectors we can construct 4 possible subset circuits using

$$\{(1,2),\ (-1,-2)\}$$
$$\{(1,1),\ (-1,-1)\}$$
$$\{(1,2),\ (1,1),\ (-2,-3)\}$$
$$\{(1,2),\ (-1,-2),\ (1,1),\ (-1,-1)\}$$

Input

Input begins with an integer $N \leq 40$ on the first line. The next N lines each contain two integer values x and y forming the vector (x, y), where $|x|, |y| \leq 10$ and $(x, y) \neq (0, 0)$. Since the given vectors are a set, all vectors are unique.

Output

Output the number of nonempty subsets of the given vectors that produce circuits. It's guaranteed that the answer is less than 10^{10}.

Sample Input 1	Sample Output 1
5 1 2 1 1 -1 -2 -2 -3 -1 -1	4