

My Treat

Time limit: 2500 ms
Memory limit: 256 MB

A group of college students has a tradition that when they eat out in a group, one of them pays for everyone's dinners. Now that the students are getting ready to graduate, they want to come up with a schedule to make sure that no one has paid for more dinners than they have received.

Consider an example where Alice has bought dinner for Bob, and Bob has bought dinner for Chuck and Dave. If Chuck buys dinner for Alice and Dave buys dinner for Bob, then everyone has paid for the same number of dinners as they have received.

For each input, you must calculate:

- The minimum number of meals that must be bought on behalf of another individual so that everyone has received the same number of dinners that they paid for.
- The minimum number of days that it will take to reach the point that no one has paid for more dinners than they have eaten, given that a person can eat at most one dinner per day. Note that a person can buy any number of dinners for others on a single day.

Standard input

The first line of input contains a single integer T , the number of test cases.

Each test case begins with an integer M , which gives the number of meals that have been occurred.

Each of the following M lines describes a meal. The first value is a string that gives the name of the person who paid for the meal. Then there is an integer N which gives the number of other people who ate at that dinner. The following N values are the names of the people whose dinner was paid for. Each of the values is separated by a space.

Standard output

For each testcase, output one line with two values, separated by a space:

- The minimum number of meals that must be bought so that everyone has received the same number of dinners that they paid for.
- The minimum number of days that it will take to reach this point.

Constraints and notes

- $1 \leq T \leq 10$
- $1 \leq M \leq 10^4$
- $1 \leq N \leq 10$

Each name will be a string made up of only upper and lower-case letters, up to 20 characters long.

Input	Output	Explanation
<pre>2 3 Alice 1 Bob Chuck 1 Dave Dave 1 Eve 4 Alice 2 Bob Chuck Chuck 1 Bob Dave 2 Alice Bob Alice 1 Eve</pre>	<pre>2 1 4 2</pre>	<p>In the sample, there are two testcases.</p> <p>In the first testcase, Alice bought Bob a dinner, Chuck bought Dave a dinner, and Dave bought Eve a dinner.</p> <p>We can achieve the objectives in a single day if Eve buys Chuck dinner and Bob buys Alice dinner. Note that it would also be ok if Eve buys Alice dinner and Bob buys Chuck dinner.</p> <p>In the second test case, Alice bought Bob and Chuck dinner, Chuck bought Bob dinner, Dave bought Alice and Bob dinner, and Alice bought Eve dinner. We can achieve zero balances with the following schedule:</p> <ul style="list-style-type: none">• Day 1: Bob buys dinner for Alice and Dave.• Day 2: Bob buys dinner for Alice, and Eve buys dinner for Dave.