



J — Jack's socks

Jack is a scientist. As you probably realize, this means he does not pay much attention to what he wears daily. He is also a man, which means that he knows the names of no more than six colors, cannot tell the difference between ecru and white and associates plum only with fruits. But today he is leaving for a conference. He has just pulled a set of single socks out of washing machine and must pair them. He can say which sock is similar to which one, and he may pair only these socks which seem similar. However, many socks are similar to many others. Worse than that, similarity relation is not necessarily transitive. For example, for Jack blue feels similar to seagreen and seagreen to green, but Jack can distinguish between blue and green and say that they are not similar.

Jack wonders if there is exactly one way to pair all his socks. Help him by writing an appropriate program. Do not worry: he might be a scientist, but he is not going to wear his socks with sandals.

Multiple Test Cases

The input contains several test cases. The first line of the input contains a positive integer $Z \leq 50$, denoting the number of test cases. Then Z test cases follow, each conforming to the format described in section *Single Instance Input*. For each test case, your program has to write an output conforming to the format described in section *Single Instance Output*.

Single Instance Input

The first line of input instance contains two integers n and m separated by a single space, where $1 \leq n \leq 1000$ and $0 \leq m \leq 10000$. Number n is even and denotes the number of socks; they are numbered from 1 to n . Each of the following m lines contains two numbers $a_i \neq b_i$ separated by a single space, which means that socks a_i and b_i are similar. Each similarity pair is listed exactly once, i.e., if (a_i, b_i) occurs, then neither (a_i, b_i) nor (b_i, a_i) appears later in this list.

Single Instance Output

For each input instance your program should check whether there exists exactly one way of pairing all these socks. If not, it should output a line containing NO. Otherwise, it should output YES in the first line, followed by $n/2$ lines, each containing a sock pair from this pairing separated by a single space. Pairs should be output in sorted order, i.e., for each pair (c, d) , it should hold that $c < d$ and for any two consecutive pairs (c, d) and (e, f) , $c < e$.

Example

| Input | Output |
|-------|--------|
| 2 | NO |
| 4 4 | YES |
| 1 2 | 1 2 |
| 2 3 | 3 4 |
| 3 4 | |
| 4 1 | |
| 4 3 | |
| 1 2 | |
| 2 3 | |
| 3 4 | |