ETH Zurich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger

HS 2015
Exercise Sheet 1

ACM Programming Challenges Lab

Exercise 1 – *Dana's dominoes*

Paul's friend Dana loves playing with dominoes. Recall that a *domino* is a rectangular tile with a line dividing one of its faces into two square ends. Each end is marked with a certain number of spots: it can either be blank (zero spots), or there can be up to $s \geq 1$ spots (in the normal game, $s = 6$). During the game, the dominoes are laid out on the table in such a way that any two adjacent dominoes have the same number of spots on the sides where they meet. This is illustrated in the figure below: the arrangement on the right is illegal because the second and third domino have different numbers of spots where they touch.

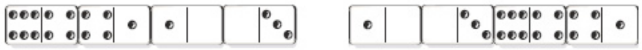

Figure 1: Legal (left) and illegal (right) arrangement of dominoes

Sometimes, between games, Dana likes to relax with a Domino puzzle of her own invention. First, she chooses an arbitrary set of domino tiles. Then, using tiles from this set, she tries to build a *domino cycle*. A domino cycle is an arrangement of dominoes on a cycle that respects the placement rule explained above, and such that that every number of spots $0, 1, \ldots, s$ appears exactly twice. A domino cycle does *not* have to use every single domino tile from the set; indeed, it is easy to see that every domino cycle contains exactly $s + 1$ dominoes.

Her only problem with this puzzle is that, sometimes, she thinks that it is not possible to build a domino cycle, but then she has no way of really being sure. Help her out by writing a program that checks if a given set of dominoes can be used to build a domino cycle.

Input The first line of the input contains the number $1 \leq T \leq 20$ of test cases. Each test case starts with a line containing two numbers n and s, separated by a space. Here, $1 \leq n \leq \binom{s+1}{2} + s + 1$ denotes the number of available dominoes, and $1 \leq s \leq 10$ denotes the maximum number of spots that can appear on a domino. This is followed by n lines of the form `u v`, where $0 \leq u, v \leq s$ denote the number of spots on the ends of a domino.

You can assume that each possible domino appears at most once in the set (counting `u v` and `v u` as the same domino). Moreover, it is possible to have dominoes of the form `u u`.

Output For every test case, your program should output, on a separate line, whether it is possible to construct a domino cycle using the given set of dominoes. Output `yes` if it is possible, and `no` otherwise.

Points There are two test sets, worth 100 points in total.

1. For the first test set, worth 30 points, you may assume that for each $0 \leq i \leq s$, you are given at most two dominoes that have i spots on one of their ends.
2. For the second test set, worth 70 points, there are no additional assumptions.

Sample Input

```
2
10 6
2 6
0 2
2 4
3 2
4 6
6 1
1 3
3 5
0 5
4 5
3 10
1 2
0 1
0 2
```

Sample Output

```
yes
no
```