

# Совершенное паросочетание

Требуется написать программу, которая с помощью вероятностного алгоритма (оценка вероятности ошибки которого основана на лемме Липтона—деМилло—Шварца—Зиппеля) проверяет, есть ли в заданном графе совершенное паросочетание. На самом деле, считать определитель необязательно: достаточно любым относительно быстрым способом проверить невырожденность соответствующей матрицы, ведь для ответа в задаче нужна только эта информация; можно воспользоваться обычным исключением Гаусса. **ВАЖНО!** Как мы обсуждали на лекции, чтобы ошибки округления/переполнения не приводили к ложному положительному ответу, нужно работать не с числами с плавающей точкой, а с **вычетами по какому-нибудь достаточно большому простому модулю**. Это обязательное требование к программе.

На вход (из стандартного потока ввода) подаётся список рёбер двудольного графа без изолированных вершин с равномоощными долями. Вершины каждой доли графа занумерованы последовательными целыми неотрицательными числами, начиная с нуля. Формат входа:

<количество рёбер>

<номер вершины из левой доли> <номер вершины из правой доли>

...

<номер вершины из левой доли> <номер вершины из правой доли>

Программа должна вывести в стандартный поток вывода единственное слово `yes`, если в графе есть совершенное паросочетание, и `no` в противном случае. Общее количество вершин графа не превосходит 200.

---

## Sample Input 1:

```
5
0 1
0 0
1 2
2 3
3 3
```

---

## Sample Output 1:

```
no
```

---

## Sample Input 2:

```
6
0 1
```

0 0  
1 2  
2 3  
3 3  
3 0

---

**Sample Output 2:**

yes