



The Islander

locked

Problem

Submissions

Leaderboard

Discussions

Известният математик Ойлер искал да мине през всички мостове на Кьонигсберг, без да минава през някой мост 2 или повече пъти. От там се ражда и известният проблем за Ойлеров цикъл и Ойлеров път, като това са обхождания на граф така, че всяко ребро се посещава точно по 1 път.

Вашата задача е по подаден граф да определите дали Ойлер може да обходи всяко ребро в графа точно по веднъж и ако да - дали накрая ще се върне във върха, от където е тръгнал, или не.

ЗАБЕЛЕЖКА: Ойлер трябва да може да посети всички върхове на графа, не е достатъчно подграф да съдържа Ойлеров цикъл/път.

Важно е да отбележим, че графът може да е мултиграф.

Input Format

Нап първият ред ще получите 1 число *queries* - броя на заявките, които ще получите.

След това *queries* пъти ще получите:

На първия ред *N M* - 2 числа, съответно броя на върховете и броя на ребрата в даден граф.

На следващите *M* реда ще получите по 2 числа *a b*, които дефинират ребро между върха *a* и върха *b* за графа описан на горния ред.

Constraints

$$1 \leq \textit{queries} \leq 100$$

$$1 \leq N, M \leq 100000$$

$$1 \leq a, b \leq 100000$$

Не е задължително върховете на графа да са индексирани с числата $[0, N - 1]$.

Output Format

За всяка една заявка изведете едно от следните 3 на нов ред:

- none, ако графът не е нито Ойлеров път, нито Ойлеров цикъл
- epath, ако графът има Ойлеров път
- ecycle, ако графът има Ойлеров цикъл

Ойлеров цикъл е по-силен от Ойлеров път. т.е ако има цикъл е гарантирано, че има и път, но изведете, че има цикъл, тъй като това дава повече информация.

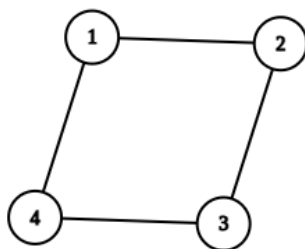
Sample Input 0

```
2
4 4
1 2
3 2
4 3
1 4
4 7
1 2
1 2
```

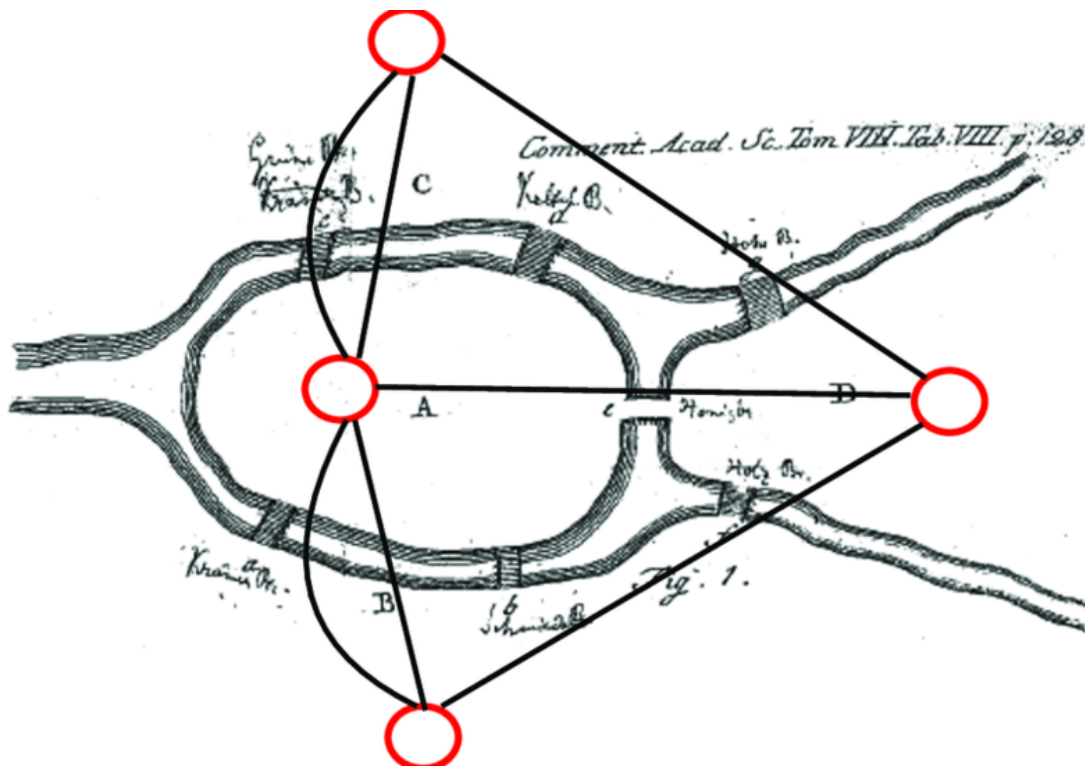
1	4
2	4
2	3
2	3
3	4

ecycle
none

Имаме 2 графа, които трябва да определим дали имат Ойлеров път/цикъл.




Първият граф има ойлеров цикъл, защото може да посетим всички ребра по един път да приключим обхождането в същия връх, от който сме тръгнали.



Вторият граф е класическия пример за Седем моста на Кьонигсберг, който за жалост не е нито Ойлеров цикъл, нито Ойлеров път.

[More](#)

Current Buffer (saved locally, editable)  

C++14   

```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 using namespace std;
7
8
9 int main() {
10     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
11     return 0;
12 }
13
```

Line: 1 Col: 1

 [Upload Code as File](#) ☐ [Test against custom input](#)[Run Code](#)[Submit Code](#)[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)