The Value of Friendship

by Voidminded

Problem | Submissions | Leaderboard | Discussions | Editorial

You're researching friendships between groups of n new college students where each student is distinctly numbered from 1 to n. At the beginning of the semester, no student knew any other student; instead, they met and formed individual friendships as the semester went on. The friendships between students are:

- *Bidirectional*. If student a is friends with student b, then student b is also friends with student a.
- *Transitive*. If student a is friends with student b and student b is friends with student c, then student a is friends with student c. In other words, two students are considered to be friends even if they are only indirectly linked through a network of mutual (i.e., directly connected) friends.

The purpose of your research is to find the maximum total value of a group's friendships, denoted by $total$. Each time a direct friendship forms between two students, you sum the number of friends that *each* of the n students has and add the sum to $total$.

You are given q queries, where each query is in the form of an unordered list of m distinct direct friendships between n students. For each query, find the maximum value of $total$ among all possible orderings of formed friendships and print it on a new line.

Input Format

The first line contains an integer, q, denoting the number of queries. The subsequent lines describe each query in the following format:

1. The first line contains two space-separated integers describing the respective values of n (the number of students) and m (the number of distinct *direct* friendships).
2. Each of the m subsequent lines contains two space-separated integers describing the respective values of x and y (where $x \neq y$) describing a friendship between student x and student y.

Constraints

- $1 \leq q \leq 16$
- $1 \leq n \leq 10^5$
- $1 \leq m \leq \min(\frac{n \cdot (n-1)}{2}, 2 \times 10^5)$

Output Format

For each query, print the maximum value of $total$ on a new line.

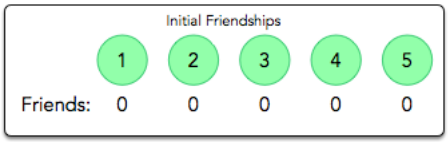
Sample Input 0

```
1
5 4
1 2
3 2
4 2
4 3
```

Sample Output 0

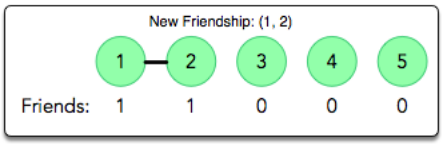
```
32
```

Explanation 0



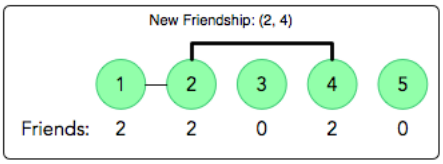
The value of $total$ is maximal if the students form the $m = 4$ direct friendships in the following order:

1. Students 1 and 2 become friends:



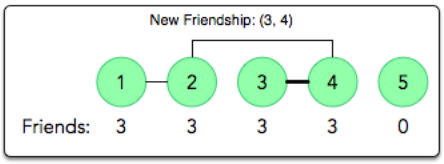
   We then sum the number of friends that each student has to get $1 + 1 + 0 + 0 + 0 = 2$.

2. Students 2 and 4 become friends:



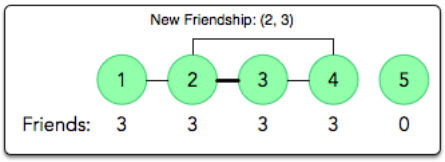
   We then sum the number of friends that each student has to get $2 + 2 + 0 + 2 + 0 = 6$.

3. Students 3 and 4 become friends:



   We then sum the number of friends that each student has to get $3 + 3 + 3 + 3 + 0 = 12$.

4. Students 3 and 2 become friends:



   We then sum the number of friends that each student has to get $3 + 3 + 3 + 3 + 0 = 12$.

When we add the sums from each step, we get $total = 2 + 6 + 12 + 12 = 32$. We then print 32 on a new line.

Submissions:243

Max Score:55
Difficulty: Hard

Rate This Challenge:

More

Current Buffer (saved locally, editable) Java 7

```
1 ▾ import java.io.*;
2   import java.util.*;
3   import java.text.*;
4   import java.math.*;
```

```java
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int t = in.nextInt();
        for(int a0 = 0; a0 < t; a0++){
            int n = in.nextInt();
            int m = in.nextInt();
            for(int a1 = 0; a1 < m; a1++){
                int x = in.nextInt();
                int y = in.nextInt();
                // your code goes here
            }
        }
    }
}

```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code