Rearrangement

N people are standing in a line. Let's denote each person in the line by an identification number between 1 and N. If we denote the location of the i-th person in the line by loc(i), the distance between two persons i and j can be denoted as |loc(i) - loc(j)|.

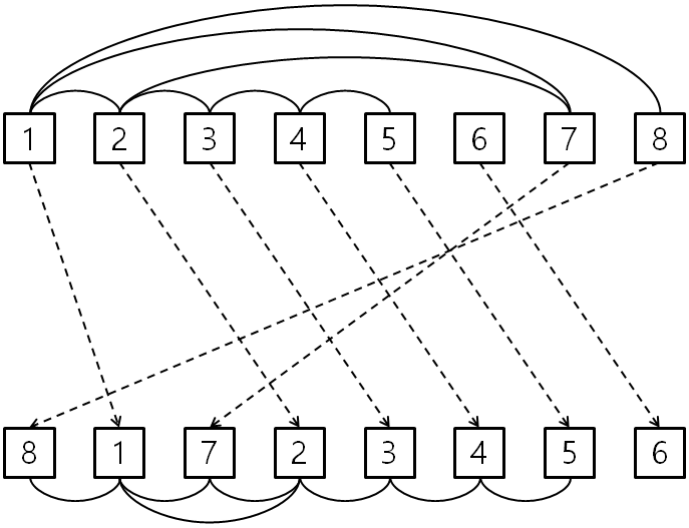
Initially, N people are standing in increasing order according to their identification numbers. In other word, for any person with ID i in the line, loc(i) = i. Some pairs of them wish to stand nearby each other. Unfortunately the initial arrangement may not allow them to do so. Considering their wishes to stand nearby for some pairs, we want to rearrange the line in order that the distance of each pair is as short as possible. The rule for rearranging the line is as follows.

> **Pick one person either from the front or from the rear of the initial line and place him(her) in the rear of the new line. Repeat this until all the persons are moved from the initial line to the new line.**

During the rearrangement process, it seems very hard for all the pairs who wish to stand nearby to place nearby in the new line. Therefore, we want to minimize the sum of the distance of every pair after rearrangement.

For instance, if we denote each pair who wishes to stand nearby in the line by a line, following figure show how a new line can be formed from the original line. Note that a curved line denotes the relationship of a pair who wishes to stand nearby and a dotted line with arrow denotes how a person moves. 8 people are standing in the initial line in increasing order according to their ID numbers. Person with ID 8 first moves from the rear of the initial line and to the first slot in the new line. Then person 1 moves to the second slot in the new line. And person 7, person 2, person 3, person 4, person 5 and person 6 move in order respectively.

This problem is a property of Special Interest Group on Theoretical Computer Science of Korean Institute of Information Scientists and Engineers. A one-time usage rights have been granted to LG Electronics.



The sum of the distances of the pairs in the original line is 22, but it becomes 8 after rearrangement and it is optimal.

[Input]

The first line of the input file contains the number T of test cases in the file. In each test case, the first line contains two integers N and M, where N denotes the number of people in the line and M the number of pairs who wish to stand nearby. In the next M lines, each line contains two numbers which denote the ID numbers of a pair.

There are two kinds of inputs listed as follows.

- Small Set: $2 \leq N \leq 10$, $1 \leq M \leq 20$
- Large Set: $2 \leq N \leq 1{,}000$, $1 \leq M \leq 20{,}000$

[Output]

For each test case given, print one line with one integer, which is the sum of distances of the pairs after rearrangement.

[I/O Example]

This problem is a property of Special Interest Group on Theoretical Computer Science of Korean Institute of Information Scientists and Engineers. A one-time usage rights have been granted to LG Electronics.

Input

```
2
8 7
1 2
3 2
3 4
4 5
7 2
1 8
1 7
4 4
1 2
2 4
1 4
3 4
```

Output

```
8
5
```

This problem is a property of Special Interest Group on Theoretical Computer Science of Korean Institute of Information Scientists and Engineers. A one-time usage rights have been granted to LG Electronics.