# Biased Standings

Usually, results of competitions are based on the scores of participants. However, we are planning a change for the next year of IPSC. During the registration each team will be able to enter a single positive integer : their preferred place in the ranklist. We would take all these preferences into account, and at the end of the competition we will simply announce a ranklist that would please all of you.

But wait... How would that ranklist look like if it won't be possible to satisfy all the requests?

Suppose that we already have a ranklist. For each team, compute the distance between their preferred place and their place in the ranklist. The sum of these distances will be called the badness of this ranklist.

## Problem specification

Given team names and their preferred placements find one ranklist with the minimal possible badness.

## Input specification

The first line of the input file contains an integer **T** specifying the number of test cases. Each test case is preceded by a blank line.

Each test case looks as follows: The first line contains **N** : the number of teams participating in the competition. Each of the next **N** lines contains a team name (a string of letters and numbers) and its preferred place (an integer between 1 and **N**, inclusive). No two team names will be equal.

## Output specification

For each of the test cases output a single line with a single integer : the badness of the best ranklist for the given teams.

## Example

**Input:**
2

7
noobz 1
llamas 2
Winn3rz 2
5thwheel 1
NotoricCoders 5
StrangeCase 7
WhoKnows 7

3
ThreeHeadedMonkey 1
MoscowSUx13 1

**Output:**
5
3

## Explanation:

In the first test case, one possible ranklist with the minimal badness is:

1. noobz
2. llamas
3. Winn3rz
4. 5thwheel
5. NotoricCoders
6. WhoKnows
7. StrangeCase

In the second test case all ranklists are equally good.

**Note: the input file will not exceed 5MB.**