Problem E
Jack Edmonds

Time limit: 2 seconds
Memory limit: 256 megabytes

Problem Description

> Jack Edmonds is an American computer science. Probably known most by the ACM-ICPC community as the co-inventor of Edmonds-Karp algorithm which solves the maximum flow problem in $O(|V||E|^2)$. However, his most important contribution to the field is perhaps Cobhan-Edmonds hypothesis which defines the concept of polynomial time as a way of telling if an algorithm is practical or not. Nowadays, we consider that the problems in P (the class of deterministic polynomial time) and the problems in NP (the class of nondeterminisitc polynomial time) are efficiently solvable and efficiently verifiable, respectively. The "P versus NP problem" is to ask whether P equals NP, and this is the major open problem in computer science and one of the seven Millennium Prize Problems. Most computer scientists believe $P \neq NP$ but they still cannot provide a correctness proof. On the other hands, many people tried to prove $P = NP$ by showing that some NP-hard problem (at least as hard as any NP problem) is in P, but they all failed.

A famous problem called the traveling salesperson problem is an example of a NP-hard problem. The problem goes as follows. Given a list of destinations and the distance between each pair of destinations, which is the shortest possible route from an origin that visits every destinations, and returns to the origin? Now, as a brilliant programmer, I'm sure you can solve this problem in no time. Only one problem though. To make the problem more challenging, we are not providing you with the map! Instead, you will be given the coordinates of each destination, and no roads have been built yet. It is obviously no way to visit all destinations if some destination is not connected to all the others.

Assume there are n destinations. Their coordinates are $(x_1, y_1), \cdots, (x_n, y_n)$, and (x_1, y_1) is the origin. The city mayor is going to help you. He will build the roads for you. However, he is stupid and lazy. Since he is stupid, he can only build a road to directly connect two destinations (x_i, y_i) and (x_j, y_j) using only horizontal and vertical segments. Therefore, its length is $|x_i - x_j| + |y_i - y_j|$. Because he is lazy, he will not build more than $n - 1$ roads for you. His consultant told him that $n - 1$ roads are enough for you to travel to all destinations. It's your call to build which roads. Can you tell me how long is the shortest route to visit every destination and return to the origin?

Input Format

There is a single integer $T (T \leq 20)$ on a line indicating the number of test cases in this problem. Next, on the first line of each test case, there is another integer $n (n \leq 100,000)$ representing the number of destinations. The next n lines contain two integers $x, y (-1,000 \leq x, y \leq 1,000)$, representing the coordinate of the destinations.

Output Format

For every test case print how long is the shortest route to visit every destination and return to the origin?

Sample Input

```
3
3
1 1
2 2
3 3
4
2 1
-1 2
-2 -1
1 -2
6
1 2
2 3
2 2
3 4
4 3
3 1
```

Sample Output

```
8
24
16
```

Problem E
Jack Edmonds

Time limit: 2 seconds
Memory limit: 256 megabytes

Problem Description

> 傑克・愛德蒙 (Jack Edmonds) 是為美國計算機科學家。或許他最為 ACM-ICPC 社群所知研究成果的是在 $O(|V||E|^2)$ 時間內解決網路最大流量問題的 Edmonds-Karp 演算法，但他最重大的影響可能是 Cobhan-Edmonds 假說：用多項式時間來定義演算法能被實踐。如今我們把決定性多項式時間 (P) 當作是可以有效率「解答」的問題類別，而非決定性多項式時間 (NP) 當作是可有效率「驗證答案」正確性的類別。所謂的 P 對 NP 問題，就是在問「這兩個問題類別，是否相等?」，這是目前計算機科學界尚未知曉答案的問題，也是七個千禧年大獎之一。多數的計算機科學家相信 $P \neq NP$，但仍缺乏正確性的證明。另一方面，許多人試著利用證明「特定 NP-hard 問題 (至少跟任一 NP 問題一樣難) 可在多項式時間內解答」來推論 $P = NP$，但也都失敗告終。

旅行推銷員問題，是一個知名的 NP-hard 問題。該問題是：「給定一個目的地清單以及任兩個目的地之間的最短距離，要如何從一個起點拜訪所有目的地，並回到起點」。我相信像你這樣優秀的程式設計失，你一定可以很快搞定這個小問題。為了讓這問題困難點，我們將不提供你地圖，你只會得到每個目的地所在的座標點，而且一開始這些目的地之間，還沒有興建任何道路。顯然的，當有一些目的地與其他目的地連通時，你無法拜訪所有的目的地。

假定有 n 個目的地，座標分別為 $(x_1, y_1), \cdots, (x_n, y_n)$ 且 (x_1, y_1) 是起點。市長願意協助你，替你興建一些道路，讓你可以拜訪所有目的地。然而，市長又笨又懶。因為笨，他只能興建由垂直與水平路段所構成，直接連通兩個目的地 (x_i, y_i)、(x_j, y_j) 的道路。這道路的長度為 $|x_i - x_j| + |y_i - y_j|$。因為懶，他只會替你興建 $n - 1$ 道路，他的顧問告訴他，$n - 1$ 條道路就足以讓你到達所有目的地。你可以決定興建哪些道路，請你告訴我從起點出發，拜訪所有目的地後回到原點的最短路徑長度是多少。

Input Format

測試輸入的第一行有一個整數 T $(T \leq 20)$ 代表有多少組測試資料。每一組測試資料的第一行，會有一整數 n $(n \leq 100,000)$ 代表有多少目的地。接下來的 n 行，每一行都會有兩個整數 x, y $(-1,000 \leq x, y \leq 1,000)$，代表對應的目的地之座標。

Output Format

對每一組測試資料，印出從起點出發，拜訪所有目的地後回到原點的最短路徑長度是多少。

Sample Input

```
3
3
1 1
2 2
3 3
4
2 1
-1 2
-2 -1
1 -2
6
1 2
2 3
2 2
3 4
4 3
3 1
```

Sample Output

```
8
24
16
```