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 $n (n \leq 10,000)$ 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
```