F - Connecting Points

Editorial (/contests/abc409/tasks/abc409_f/editorial)

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 500 points

Problem Statement

There is a graph G with N vertices and 0 edges on a 2-dimensional plane. The vertices are numbered from 1 to N, and vertex i is located at coordinates (x_i, y_i).

For vertices u and v of G, the distance $d(u, v)$ between u and v is defined as the Manhattan distance $d(u, v) = |x_u - x_v| + |y_u - y_v|$.

Also, for two connected components A and B of G, let $V(A)$ and $V(B)$ be the vertex sets of A and B, respectively. The distance $d(A, B)$ between A and B is defined as $d(A, B) = \min\{d(u, v) \mid u \in V(A), v \in V(B)\}$.

Process Q queries as described below. Each query is one of the following three types:

- `1 a b` : Let n be the number of vertices in G. Add vertex $n + 1$ to G with coordinates $(x_{n+1}, y_{n+1}) = (a, b)$.
- `2` : Let n be the number of vertices in G and m be the number of connected components.
  - If $m = 1$, output `-1`.
  - If $m \geq 2$, merge all connected components with the minimum distance and output the value of that minimum distance. Formally, let the connected components of G be $A_1, A_2, \dots, A_m$ and let $k = \min_{1 \leq i < j \leq m} d(A_i, A_j)$. For all pairs of vertices (u, v) $(1 \leq u < v \leq n)$ that are not in the same connected

component and satisfy $d(u, v) = k$, add an edge between vertices u and v. Then, output k.

- `3 u v` : If vertices u and v are in the same connected component, output `Yes`; otherwise, output `No`.

Constraints

- $2 \leq N \leq 1500$
- $1 \leq Q \leq 1500$
- $0 \leq x_i, y_i \leq 10^9$
- For queries of type $1, 0 \leq a, b \leq 10^9$.
- For queries of type 3, let n be the number of vertices in G just before processing that query, then $1 \leq u < v \leq n$.
- All input values are integers.

Input

The input is given from Standard Input in the following format, where query_i is the i-th query to be processed.

$$
\begin{array}{l}
N \quad Q \\
x_1 \quad y_1 \\
x_2 \quad y_2 \\
\vdots \\
x_N \quad y_N \\
\mathrm{query}_1 \\
\mathrm{query}_2 \\
\vdots \\
\mathrm{query}_Q
\end{array}
$$

Each query is given in one of the following three formats:

$$1 \quad a \quad b$$

$$2$$

$$3 \quad u \quad v$$

Output

Output the answers to the queries separated by newlines, following the instructions in the problem statement.

Sample Input 1

Copy

Copy

```
4 11
3 4
3 3
7 3
2 2
3 1 2
2
3 1 2
1 6 4
2
3 2 5
2
3 2 5
2
1 2 2
2
```

Sample Output 1

Copy

Copy

```
No
1
Yes
2
No
3
Yes
-1
0
```

Initially, vertices $1, 2, 3, 4$ are located at coordinates $(3, 4), (3, 3), (7, 3), (2, 2)$, respectively.

- For the 1st query, vertices 1 and 2 are not connected, so output `No`.
- For the 2nd query, there are 4 connected components, and the vertex set of each connected component is $\{1\}, \{2\}, \{3\}, \{4\}$. The minimum distance between different connected components is 1, and an edge is added between vertices 1 and 2. Output 1.
- For the 3rd query, vertices 1 and 2 are connected, so output `Yes`.
- For the 4th query, add vertex 5 at coordinates $(6, 4)$.
- For the 5th query, there are 4 connected components, and the vertex set of each connected component is $\{1, 2\}, \{3\}, \{4\}, \{5\}$. The minimum distance between

different connected components is 2, and edges are added between vertices 2 and 4 and between vertices 3 and 5. Output 2.
- For the 6th query, vertices 2 and 5 are not connected, so output `No`.
- For the 7th query, there are 2 connected components, and the vertex set of each connected component is $\{1, 2, 4\}, \{3, 5\}$. The minimum distance between different connected components is 3, and an edge is added between vertices 1 and 5. Output 3.
- For the 8th query, vertices 2 and 5 are connected, so output `Yes`.
- For the 9th query, there is 1 connected component, so output -1.
- For the 10th query, add vertex 6 at coordinates $(2, 2)$.
- For the 11th query, there are 2 connected components, and the vertex set of each connected component is $\{1, 2, 3, 4, 5\}, \{6\}$. The minimum distance between different connected components is 0, and an edge is added between vertices 4 and 6. Output 0.
