Graph Representation in Programming Assignments

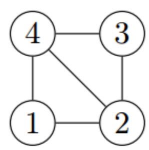
In programming assignments, graphs are given as follows. The first line contains non-negative integers n and m – the number of vertices and the number of edges respectively. The vertices are always numbered from 1 to n. Each of the following m lines defines an edge in the format u v where $1 \leq u, v \leq n$ are endpoints of the edge. If the problem deals with an undirected graph this defines an undirected edge between u and v, In case of a directed graph this defines a directed edge from u to v. If the problem deals with a weighted graph then each edge is given as u v w where u and v are vertices and w is a weight.

It is guaranteed that a given graph is simple. That is, it does not contain self-loops (edge going from a vertex to itself) and parallel edges.

Examples:

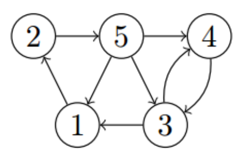
- An undirected graph with four vertices and five edges:

```
4 5
2 1
4 3
1 4
2 4
3 2
```



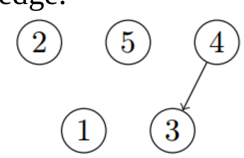
- A directed graph with five vertices and eight edges:

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



- A directed graph with five vertices and one edge:

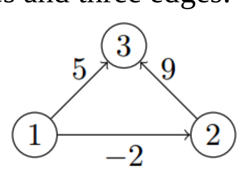
```
5 1
4 3
```



Note that the vertices $1, 2,$ and 5 are isolated (have no adjacent edges), but they are still present in the graph.

- A weighted directed graph with three vertices and three edges:

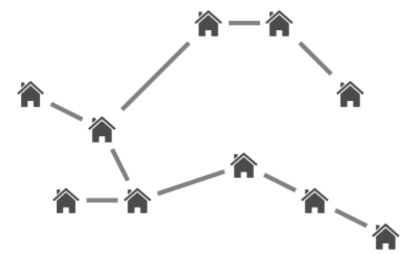
```
3 3
2 3 9
1 3 5
1 2 -2
```



1 Building Roads to Connect Cities

Problem Introduction

In this problem, the goal is to build roads between some pairs of the given cities such that there is a path between any two cities and the total length of the roads is minimized.



Problem Description

Task. Given n points on a plane, connect them with segments of minimum total length such that there is a path between any two points. Recall that the length of a segment with endpoints (x_1, y_1) and (x_2, y_2) is equal to $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

Input Format. The first line contains the number n of points. Each of the following n lines defines a point (x_i, y_i).

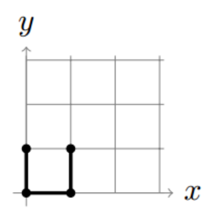
Constraints. $1 \leq n \leq 200$; $-10^3 \leq x_i, y_i \leq 10^3$ are integers. All points are pairwise different, no three points lie on the same line.

Output Format. Output the minimum total length of segments. The absolute value of the difference between the answer of your program and the optimal value should be at most -10^6. To ensure this, output your answer with at least seven digits after decimal point (otherwise your answer, while being computed correctly, can turn out to be wrong because of rounding issues).

Sample 1.

Input:

```
4
0 0
0 1
1 0
1 1
```



Output:

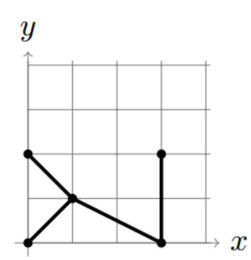
```
3.0000000
```

An optimal way to connect these four points is shown above. Note that there exists other ways of connecting there points by segments of total weight 3.

Sample 2.

Input:

```
5
0 0
0 2
1 1
3 0
3 2
```



Output:

```
7.0644951
```

An optimal way to connect these five points is shown above. The total length here is $2\sqrt{2} + \sqrt{5} + 2$.

Starter Files

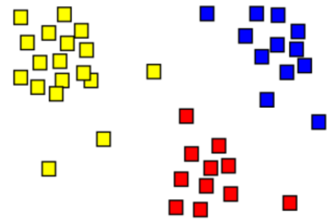
The starter solutions for this problem read the input data from the standard input, pass it to a blank procedure, and then write the result to the standard output. You are supposed to implement your algorithm in this blank procedure if you are using `C++`, `Java`, or `Python3`. For other programming languages, you need to implement a solution from scratch.

What To Do

To solve this problem, it is enough to implement carefully the corresponding algorithm covered in the lectures.

2 Clustering

Problem Introduction



Clustering is a fundamental problem in data mining. The goal is to partition a given set of objects into subsets (clusters) in such a way that any two objects from the same subset are close (or similar) to each other, while any two objects from different subsets are far apart.

Problem Description

Task. Given n points on a plane and an integer k, compute the largest possible value of d such that the given points can be partitioned into k non-empty subsets in such a way that the distance between any two points from different subsets is at least d.

Input Format. The first line contains the number n of points. Each of the following n lines defines a point (x_i, y_i). The last line contains the number k of clusters.

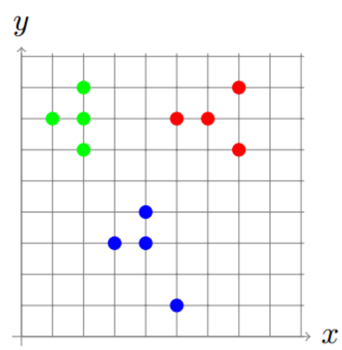
Constraints. $2 \le k \le n \le 200$; $-10^3 \le x_i, y_i \le 10^3$ are integers. All points are pairwise different.

Output Format. Output the largest value of d. The absolute value of the difference between the answer of your program and the optimal value should be at most -10^6. To ensure this, output your answer with at least seven digits after decimal point (otherwise your answer, while being computed correctly, can turn out to be wrong because of rounding issues).

Sample 1.

Input:

```
12
7 6
4 3
5 1
1 7
2 7
5 7
3 3
7 8
2 8
4 4
6 7
2 6
3
```



Output:

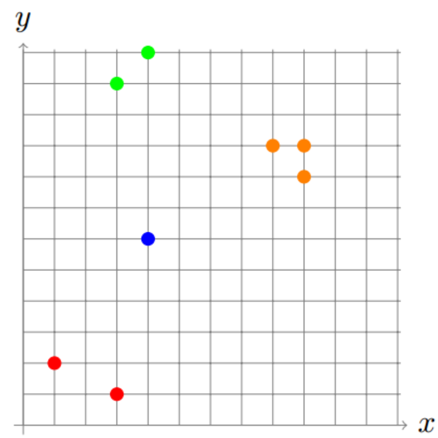
```
2.8284271
```

The answer is $\sqrt{8}$. The corresponding partition of the set of points into three clusters is shown above.

Sample 2.

Input:

```
8
3 1
1 2
4 6
9 8
9 9
8 9
3 11
4 12
4
```



Output:

```
5.0000000
```

The answer is 5. The corresponding partition of the set of points into four clusters is shown above.

Starter Files

The starter solutions for this problem read the input data from the standard input, pass it to a blank procedure, and then write the result to the standard output. You are supposed to implement your algorithm in this blank procedure if you are using `C++`, `Java`, or `Python3`. For other programming languages, you need to implement a solution from scratch.

What To Do

Think about ways of adapting the Kruskal's algorithm for solving this problem.