Programming Assignment 5: Minimum Spanning Trees

Revision: June 19, 2018

Introduction

Welcome to the third (and the last one) programming assignment of the Graphs Algorithms course! In this programming assignment you will be practicing implementing algorithms computing minimum spanning trees.

Learning Outcomes

Upon completing this programming assignment you will be able to:

1. connect the given cities by roads of minimum total length such that there is a path between any two cities;
2. compute an optimal clustering of the given set of objects.

Passing Criteria: 1 out of 2

Passing this programming assignment requires passing at least 1 out of 2 programming challenges from this assignment. In turn, passing a programming challenge requires implementing a solution that passes all the tests for this problem in the grader and does so under the time and memory limits specified in the problem statement.

Contents

Graph Representation in Programming Assignments

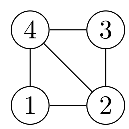
In programming assignments, graphs are given as follows. The first line contains non-negative integers n and m — the number of vertices and the number of edges respectively. The vertices are always numbered from 1 to n. Each of the following m lines defines an edge in the format `u v` where $1 \leq u, v \leq n$ 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 (edges going from a vertex to itself) and parallel edges.

Examples:

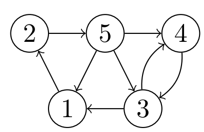
- An undirected graph with four vertices and five edges:

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



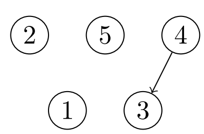
- A directed graph with five vertices and eight edges.

```
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.

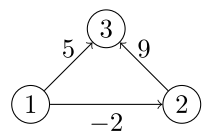
```
5 1
4 3
```



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

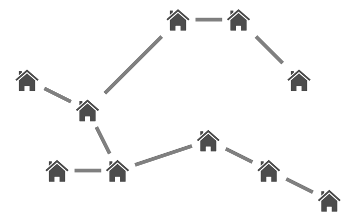
- A weighted directed graph with three vertices and three edges.

```
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. $1 \le n \le 200$; $-10^3 \le x_i, y_i \le 10^3$ 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 the decimal point (otherwise your answer, while being computed correctly, can turn out to be wrong because of rounding issues).

Time Limits.

language	`C`	`C++`	`Java`	`Python`	`Haskell`	`JavaScript`	`Scala`
time (sec)	2	2	3	10	4	10	6

512

Sample 1.

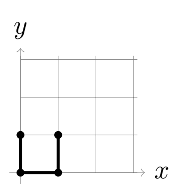
Input:

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

Output:

```
3.000000000
```

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



Sample 2.

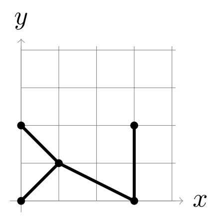
Input:

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

Output:

```
7.064495102
```

An optimal way to connect these four points is shown below.



The total length here is equal to $2\sqrt{2} + \sqrt{5} + 2$.

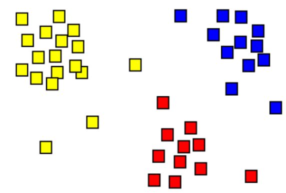
Need Help?

Ask a question or check out the questions asked by other learners at this forum thread.

2 Clustering

Problem Introduction

Clustering is a fundamental problem in data mining. The goal is to partition a given set of objects into subsets (or 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. $2 \leq k \leq n \leq 200$; $-10^3 \leq x_i, y_i \leq 10^3$ 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 the decimal point (otherwise your answer, while being computed correctly, can turn out to be wrong because of rounding issues).

Time Limits.

language	C	C++	Java	Python	Haskell	JavaScript	Scala
time (sec)	2	2	3	10	4	10	6

512

Sample 1.

Input:

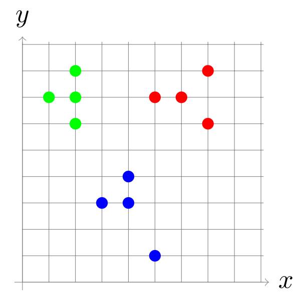
```
12
7 6
4 3
5 1
1 7
2 7
5 7
3 3
7 8
2 8
4 4
6 7
2 6
3
```

Output:

```
2.828427124746
```

Explanation:

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



Sample 2.

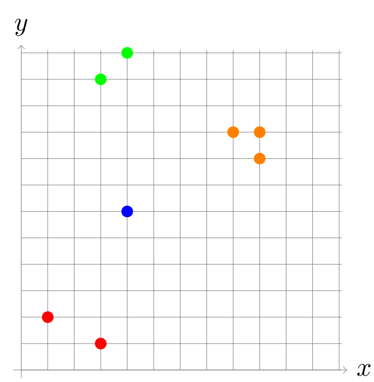
Input:

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

Output:

```
5.000000000
```

Explanation:
The answer is 5. The corresponding partition of the set of points into four clusters is shown below.



What To Do

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

Need Help?

Ask a question or check out the questions asked by other learners at this forum thread.