

# CE City Connections CODE

10 Possible Points

| 12/5/2022

Attempt 1



**IN PROGRESS**

Next Up: Submit Assignment



Add Comment

Unlimited Attempts Allowed

▼ Details

Graphs

## CE: City Connections CODE



### Learning Objectives

- Determine the shortest way to connect all vertices in a given graph.
- Create a new class by modifying existing code and adapting it to similar but different requirements.



### Overview

In this CE, you will determine how to connect all the cities in a given graph with bike trails while minimizing the miles of bike trails that need to be built.

◀ [Previous](#)

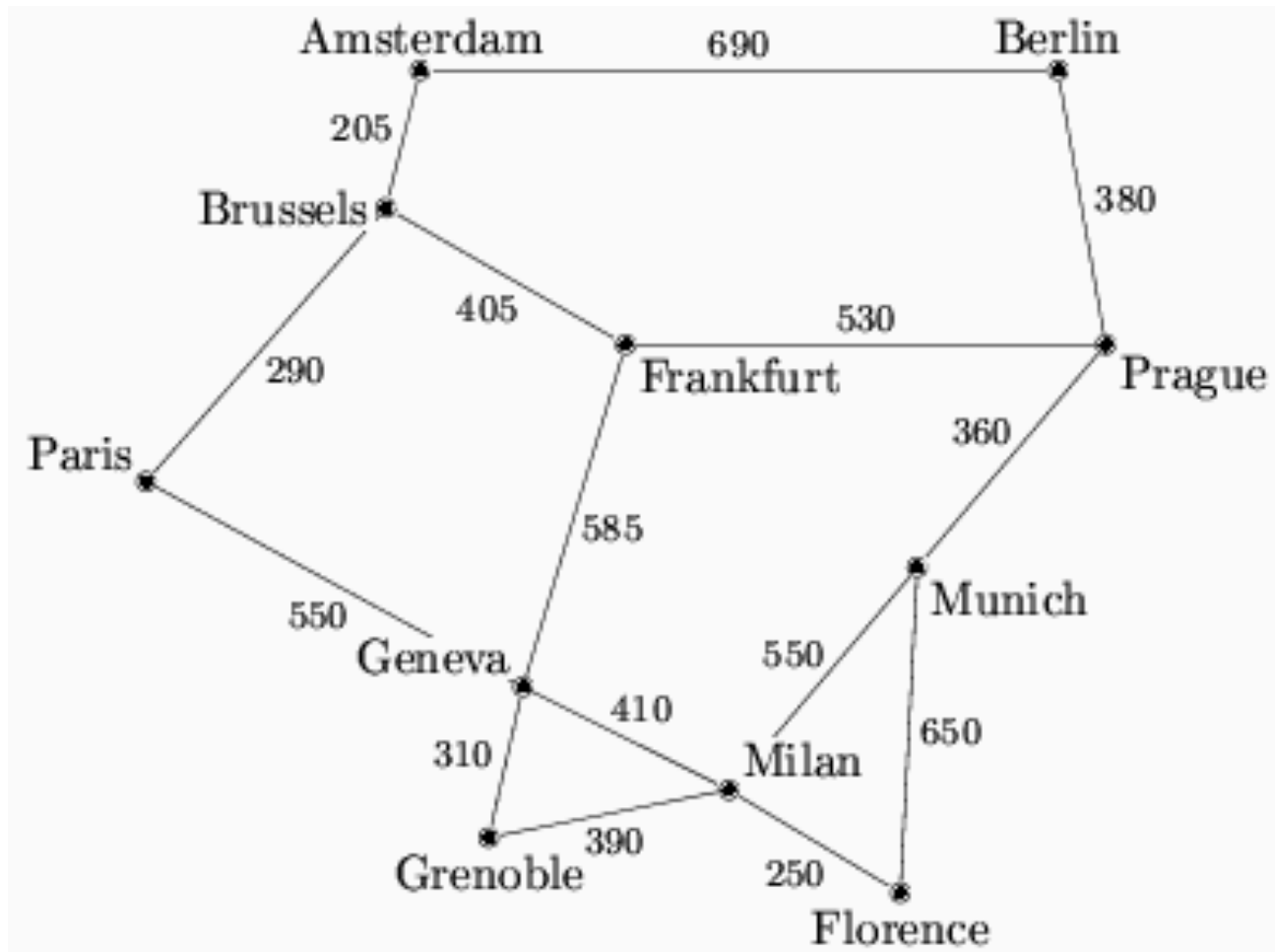
(<https://slcc.instructure.com/courses/817632/modules/items/18753107>)

Submit Assignment

[Next](#) ▶

(<https://slcc.instructure.com/courses/817632/modules/items/18753110>)

The following graph shows European cities and the distances between those cities in km.



Here is a CSV file that includes the data from the graph above:

[CityConnections.csv \(https://slcc.instructure.com/courses/817632/files/135713676/download?wrap=1\)](https://slcc.instructure.com/courses/817632/files/135713676/download?wrap=1) [↓ \(https://slcc.instructure.com/courses/817632/files/135713676/download?download\\_frd=1\)](https://slcc.instructure.com/courses/817632/files/135713676/download?download_frd=1)

Let's assume the European Union wants to enable its people to reach all of those cities via bike trail.

Because there is a limited budget, you are tasked to find the shortest way to connect all the cities

[< Previous](https://slcc.instructure.com/courses/817632/modules/items/18753107)

[\(https://slcc.instructure.com/courses/817632/modules/items/18753107\)](https://slcc.instructure.com/courses/817632/modules/items/18753107)

Submit Assignment

[Next >](https://slcc.instructure.com/courses/817632/modules/items/18753110)

[\(https://slcc.instructure.com/courses/817632/modules/items/18753110\)](https://slcc.instructure.com/courses/817632/modules/items/18753110)

[/SymbolGraph.java.html](#)) (<https://algs4.cs.princeton.edu/42digraph>

[/SymbolDigraph.java.html](#)), however, it creates an edge-weighted graph rather than a graph.

The class `EdgeWeightedSymbolGraph` should provide the public methods `contains`, `indexOf`, and `nameOf`. It also includes a method named `graph`. However, its return type should be `EdgeWeightedGraph`, not `Graph`.

Expected Graph File:

The provided graph file also has a different format. Each line includes the names of two connected vertices (as strings) followed by the weight. The vertices and the weight are separated by a delimiter that is provided as an argument to the constructor. In our case, the delimiter is a comma.

Even though the weights of this example happen to be whole numbers, the resulting `EdgeWeightedGraph` should be able to accept weights that are floating point numbers (double)

- Create a class called **CityConnectionsCE**. It includes the main method.  
Dynamically, determine the shortest way to connect all the cities with bike trails.
- Print the following:
  - A list of all the pairs of cities, that need to be connected.  
Each pair should be printed in a separate row where two cities are separated by a space.
  - The total distance of the bike trail that connects all the cities

*Important:*

*Find a solution that solves this problem not just for this specific graph but for any graph that is provided in the same format. In other words: If I add or remove rows from the CSV file, the output needs to change accordingly.*



## Expected Output

Cities to connect with a bike trail:

[< Previous](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753107>

Submit Assignment

[Next >](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753110>

Geneva to Grenoble (310.0)  
Grenoble to Milan (390.0)  
Milan to Munich (550.0)  
Milan to Florence (250.0)  
Munich to Prague (360.0)  
Prague to Frankfurt (530.0)

Total length of the bike trail: 3670.0



## Submission

Create a screen recording following the [guidelines for lab recordings](https://slcc.instructure.com/courses/817632/pages/guidelines-for-ce-recordings) (<https://slcc.instructure.com/courses/817632/pages/guidelines-for-ce-recordings>).

The video should be **30 - 60 seconds** long.

Post the video.

 (<https://slcc.instructure.com/courses/817632/modules/items/18753100>)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753101>)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753102>)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753103>)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753104>)

 ([https://slcc.instructure.com/api/v1/courses/817632/module\\_item\\_redirect/18753106](https://slcc.instructure.com/api/v1/courses/817632/module_item_redirect/18753106))

 ([https://slcc.instructure.com/api/v1/courses/817632/module\\_item\\_redirect/18753107](https://slcc.instructure.com/api/v1/courses/817632/module_item_redirect/18753107))

 (<https://slcc.instructure.com/courses/817632/modules/items/18753109>)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753110>)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753111>)

[Previous](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753107>

Submit Assignment

[Next](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753110>

**Choose a submission type****T**

Text



Upload



Office 365



More

< Previous

**(<https://slcc.instructure.com/courses/817632/modules/items/18753107>)**

Submit Assignment

Next >

**(<https://slcc.instructure.com/courses/817632/modules/items/18753110>)**