Final CODE

12/13/2023

50 Possible Points

Attempt 1

In Progress
NEXT UP: Submit Assignment

Add Comment

Unlimited Attempts Allowed
12/13/2023 to 12/13/2023

Details

Important Information:

1. **Permitted Resources:**
   Feel free to use books, notes, your code from class exercises.
   You can use the internet - but only to look up information that was posted independently of the module test.
2. **Complete the challenge on your own.**
   In other words: **no** communication or collaboration - not with humans and not with bots / AI.
   One exception: feel free to reach out to your instructor for clarifications on what to do.
3. **Don't share** the instructions or your solution - not during and not after the module test.
4. **No late points**
   The difference between the due date and until-date on this module test is a technicality.
   Ensure to submit before the until-date; there will be **no** late points.

Getting Started | Style | Best Practices: (max 5 points)

- Create a new Java project called **Final_2420**
- Include a package called **finalCode**.
- Add the following class: **FinalChallenges** It should include a main method.
- Add a folder inside the package finalCode. It should be called **Resources**.
  This is the folder where the graph file(s) will be stored.
- Ensure that each Java file includes your name in the doc comment of the class.
- Add the file **algs4.jar** to the build path.

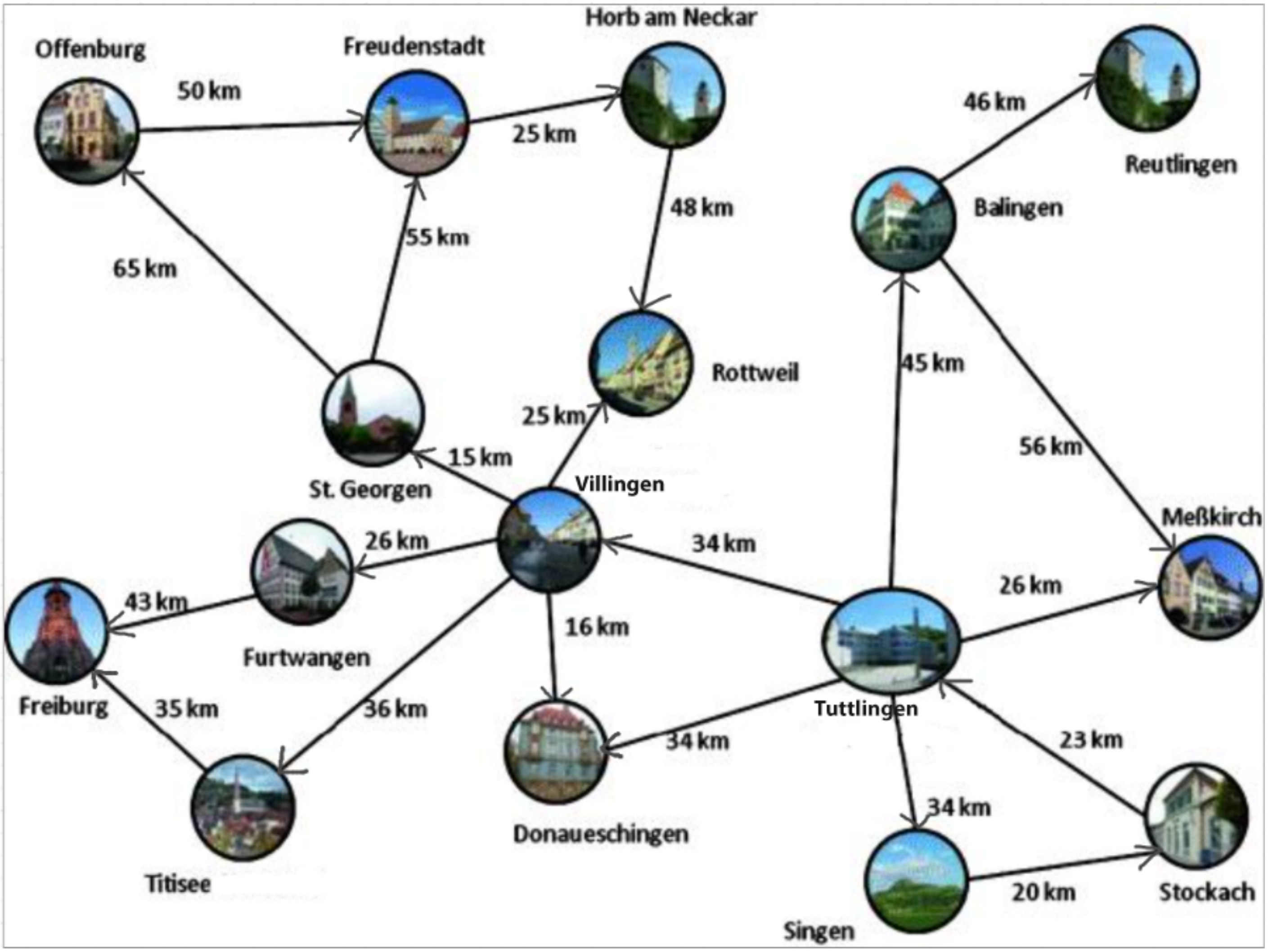
Previous
(https://slcc.instructure.com/courses/915963/modules/items/21581720)

Submit Assignment

FinalChallenges:

Important:

1. Use only classes from algs4, java.lang, and/or one of the symbol graph adaptations we wrote as part of the CEs.
2. Avoid hardcoding.
   If I change the router names or add/remove vertices and edges in the graph file, the output needs to change accordingly.



Graph:

- Each **vertex** represents a city.
- Each **edge** indicates that the city offers roundtrips to the specified destination.

Previous

(https://slcc.instructure.com/courses/915963/modules/items/21581720)

Submit Assignment

Helper Files:

To avoid tedious typing, I provide a CSV file that includes all the connections between cities, followed by their distance.
Cities.csv (https://slcc.instructure.com/courses/915963/files/155820499?wrap=1) ↓ (https://slcc.instructure.com/courses/915963/files/155820499/download?download_frd=1) (https://slcc.instructure.com/courses/650718/files/110424000/download?download_frd=1)

I also provide the modified symbol graph classes we worked on in class, in case you need a running version.
EdgeWeightedSymbolGraph.java (https://slcc.instructure.com/courses/915963/files/155820496?wrap=1) ↓ (https://slcc.instructure.com/courses/915963/files/155820496/download?download_frd=1) EdgeWeightedSymbolDigraph.java (https://slcc.instructure.com/courses/915963/files/155820495?wrap=1) ↓ (https://slcc.instructure.com/courses/915963/files/155820495/download?download_frd=1)

(https://slcc.instructure.com/courses/650718/files/110424068/download?download_frd=1)

Challenge 1: (max 24 points)

Format the output to match the expected output (1 out of 24 points)

- Print a title (1 out of 24 points)
- Read in a city from the user (2 out of 24 points)
- If the user enters a city that is not part of the graph (4 out of 24 points)
  print a message of the following format:
  {city} is not part of the graph.
  Where {city} is replaced by the name the user entered.
- Print all the roundtrips that can be made from the specified city.
  For each roundtrip, list the destination and the kilometers the bus needs to drive for the roundtrip.
  (8 out of 24 points) for listing the neighbors
  (4 out of 24 points) for listing the names of the neighbors
  (4 out of 24 points) for listing the correct distance the bus needs to drive for the roundtrip

Expected Output:

The outputs below show the expected results when running challenge1 multiple times and providing

Output1 (user input is marked in green)

```
Challenge 1
===========
City: Balingen
Roundtrips from Balingen:
Reutlingen 92.0 km
Messkirch 112.0 km
```

Output2

```
Challenge 1
===========
City: Freiburg
Roundtrips from Freiburg:
```

Output3

```
Challenge 1
===========
City: Moab
Moab is not part of the graph.
```

Challenge 2: (max 20 points)

The old power lines that provided electricity to the cities for decades are outdated and need to be replaced with underground cables. Installing underground cables is expensive, and the cost is proportional to the distance.

- Print a title (1 out of 20 points)
- Determine how to connect all the cities with underground cables in the most cost-effective way.
  Print all the cities that need to be connected with an underground cable.
  (8 out of 20 points) for listing the right cities that need to be connected
  (4 out of 20 points) for printing the corresponding names of the cities when printing the legs.
- Assuming that one kilometer of underground cable costs 1.2 million dollars, how much does it cost to connect all cities most cost-effectively? (6 out of 20 points)
- Format the output as shown in the expected output (1 out of 20 points)
  The order in which the connections are listed may vary.

< Previous
(https://slcc.instructure.com/courses/915963/modules/items/21581720)

Submit Assignment

```
Villingen to Donaueschingen
Singen to Stockach
Stockach to Tuttlingen
Villingen to Rottweil
Freudenstadt to HorbAmNeckar
Tuttlingen to Messkirch
Villingen to Furtwangen
Tuttlingen to Villingen
Titisee to Freiburg
Villingen to Titisee
Tuttlingen to Balingen
Balingen to Reutlingen
HorbAmNeckar to Rottweil
Offenburg to Freudenstadt

Cost to connect all cities with underground cable: $564.0 million
```

Submission:

When you are done, submit the java files.
Alternatively, you can create a ZIP file of your project and submit that.

Good Luck!

View Rubric

Previous
(https://slcc.instructure.com/courses/915963/modules/items/21581720)

Submit Assignment

Some Rubric

Criteria	Ratings	Pts
Getting Started | Style | Best Practices view longer description		/ 7 pts
Challenge A view longer description		/ 15 pts
Challenge B view longer description		/ 14 pts
Challenge C view longer description		/ 14 pts
		Total Points: 0

Text Upload Office 365 More

Previous

(https://slcc.instructure.com/courses/915963/modules/items/21581720)

Submit Assignment