

# Graph Routes, Lab Assignment 15

**Due** Dec 13 by 11:59pm **Points** 100 **Submitting** a file upload **File Types** cpp

## Part 1

### Write A Function For Shortest Route [ GraphShortest.cpp ]

Write **GraphShortest.cpp**, by completing the supplied CPP file. Click [GraphShortest.cpp](#)  download the supplied file that contains the main function and the function for shortest route. Click [cities.txt](#)  to download the input file to use, containing road map information for California. Write the body for the missing shortest route function, based on the algorithm in the lecture notes.

Note that the adjacency list is STL pairs of ints and doubles. Traverse it with a `list<pair<int, double> >::iterator`. Then use the `.first` and `.second` public data members of the pair to access the neighbor's index and the cost of getting there.

Submit the CPP to the class website for credit. Do NOT submit the TXT file.

**Program I/O.** Input: As prompted, a start city and a destination city. Output: The shortest calculated routes from start to destination.

**Program I/O.** Here's what the result should be for Coos Bay to Yuma:

```
Enter the source city [blank to exit]: Coos Bay
Enter the destination city [blank to exit]: Yuma
Total edges: 5-Coos Bay-Eureka-San Francisco-Bakersfield-Los Angeles-Yuma

Enter the source city [blank to exit]: San Francisco
Enter the destination city [blank to exit]: Sacramento
Total edges: 1-San Francisco-Sacramento

Enter the source city [blank to exit]:
Done!
```

## Part 2

### Write A Function For Cheapest Route [ GraphCheapest.cpp ]

Write **GraphCheapest.cpp**, by modifying the CPP file for the shortest route. Modify your CPP from part 1, replacing the shortest route function with a cheapest route function, using Dijkstra's algorithm. Add any struct definition(s) as needed.

Submit the CPP to the class website for credit. Do NOT submit the TXT file.

**Program I/O.** Input: Same as part 1 Output: Same as part 1, except for the cheapest route, with #of miles outputted instead of #of edges

**Program I/O.** Here's what the result should be for San Francisco to Reno:

```
Enter the source city [blank to exit]: San Francisco
Enter the destination city [blank to exit]: Reno
Total miles: 229-San Francisco-Sacramento-Reno

Enter the source city [blank to exit]: Yuma
Enter the destination city [blank to exit]: Coos Bay
Total miles: 1164-Yuma-Los Angeles-Bakersfield-San Francisco-Eureka-Coos Bay

Enter the source city [blank to exit]:
Done!
```

**Lab Assignment Rubric**

Criteria	Ratings								Pts
Fully accurate results, following all specifications <a href="#">view longer description</a>	Works the first time. 70.0 pts	Works on the 2nd try 65.0 pts	Works on the 3rd try 60.0 pts	Works after 4 or more tries. 50.0 pts	Doesn't work after 2 weeks. Partial credit. 20.0 pts	Not submitted within two weeks of the due date. 0.0 pts	Work is not original -- appears to be a marked-up copy of the work of another or previous student. 0.0 pts		70.0 pts
Submits all work on time, fully complete if not fully correct. <a href="#">view longer description</a>	Submitted on time 20.0 pts	Submitted on time, but one or more files are missing or not correctly named. 16.0 pts			Submitted on time, but with missing identification in one or more submitted CPP or H files. 15.0 pts		Submitted on time but not fully complete. 10.0 pts	Late or wholly incomplete! 0.0 pts	20.0 pts
Well-organized and professional quality code. <a href="#">view longer description</a>	Fully meets expectations 10.0 pts	Mostly meets expectations, just needs to be a bit more careful. 8.0 pts			Many areas are well done, but there are a lot of areas that need work. 6.0 pts		Getting there, but needs to be a lot better. 3.0 pts	Needs a lot of work. See the instructor for guidance. 0.0 pts	10.0 pts
Total Points: 100.0									