Tesla Autopilot C++ Coding Challenge

Your objective is to construct a search algorithm to find the minimum time path through the tesla network of supercharging stations. Each supercharger will refuel the vehicle at a different rate given in km/hr of charge time. Your route does not have to fully charge at every visited charger, so long as it never runs out of charge between two chargers. **You should expect to need no more than 4 hours to solve this problem**. We suggest implementing a quick brute force method before attempting to find an optimal routine.

You will be provided with a code skeleton which includes a header with the charger network data in the format: name, latitude in degrees, longitude in degrees, charge rate in km/hr

You may compare your solutions against our reference implementation using the provided "checker" programs in either OSX or linux; make sure to use it to check your submission Input: Your program should take as input two strings: “start charger name”, “end charger name"

Output: Your program’s only output should be a print to std::out of a string in the format: initial charger name, first charger name, charge time in hrs, second charger name, charge time in hrs, …, …, goal charger name. **This is the format required by the checker program** as well.

For example the command ./solution Council_Bluffs_IA Cadillac_MI

might return:

Council_Bluffs_IA, Worthington_MN, 1.18646, Albert_Lea_MN, 1.90293, Onalaska_WI,
0.69868, Mauston_WI, 1.34287, Sheboygan_WI, 1.69072, Cadillac_MI

You may make the following assumptions:

- The car begins at the start charger with a full charge of 320km
- The car travels at a constant speed of 105km/hr along great circle routes between chargers
- The Earth is a sphere of radius 6356.752km

Your submission will be evaluated in terms of the following metrics:

- Path satisfiability (car remains above 0km of range throughout entire trip)
- Path optimality (total time driving + charging)
- Coding structure
- Code clarity
- Computational cost

You should ensure that your submission compiles under g++ optimization level 1 in a standard linux environment, for example:

```
g++ -std=c++11 -O1 main.cpp network.cpp -o candidate_solution
```

If your solution includes additional cpp files, please include a readme file with the appropriate compiler string.