README.md

Github Organization:
The bold part is for grading.

folders
> **data**

  — “data_edge_617.csv”,
  — “data_vertex_262.csv”
  as our input file.
> **src**

  — bfs_traversal.cpp
  — brandes_algorithm.cpp
  — export_bc.cpp
  — fruchterman_Reingold.cpp
  — get_functions.cpp
  — graph.hpp
all the code
> tests

  > test_data (all the test data input)

  — catch.hpp
  — tests.cpp
Betweenness_Centrality_Table.csv (the result of brandes algorithm)
main.cpp
Makefile
data_cleaning.rmd
Project contract

Result.md
ReadMe.md
cs225_final_slides.ppt

Presentation video:
> https://mediaspace.illinois.edu/media/t/1_wcm6edgz

Presentation slide:
> **cs225_final_slides.ppt**

Running Instructions:

Run the code

```
make
./main
```

To get the “Betweenness_Centrality_Table.csv” and “” Graph.
Note: It usually takes 30 seconds.

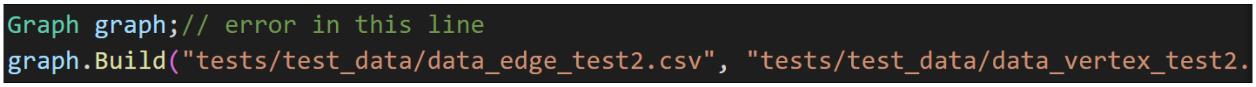
Test the algorithm

```
make tests
./tests/tests
```

We have 3 test cases for the ConnectedComponents() function, which is based on the BFS().
We also have 3 test cases for Brandes() function, which is based on the BFS4ST().
There are a total of 13 assertions in 6 test cases.
Note: It usually takes 15 seconds.

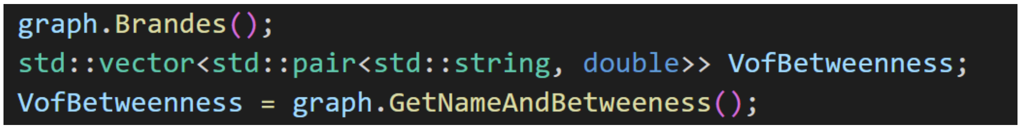
Important functions:

Several functions are called in the main.cpp:



```
Graph graph;// error in this line
graph.Build("tests/test_data/data_edge_test2.csv", "tests/test_data/data_vertex_test2.
```

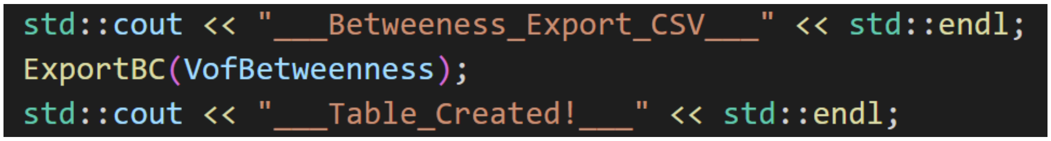
1. Graph graph -> Create an object of My Class (Graph)
2. graph.Build(“tests/test_data/data_edge_test2.csv”, “tests/test_data/data_vertex_test2.csv”)
   The inputs are a “edge.csv”, a “vertex.csv”.

   ```
   graph.Brandes();
   std::vector<std::pair<std::string, double>> VofBetweenness;
   VofBetweenness = graph.GetNameAndBetweeness();
   ```

3. graph.Brandes()
   Run the Brandes algorithm to get VofBetweeness, which is a vector of betweenness centrality.
4. graph.GetNameAndBetweeness()

Get the result, which is a vector of std::pair<string,double>. The string is the name of the school, and the double is the betweenness centrality of that vertex (school).



```
std::cout << "___Betweeness_Export_CSV___" << std::endl;
ExportBC(VofBetweenness);
std::cout << "___Table_Created!___" << std::endl;
```

5. ExportBC(VofBetweenness)
   write the **Betweenness_Centrality_Table.csv.**
6. graph.updateCC();
7. std::vector<std::vector<Node>> cc = graph.GetConnectedComponents();
8. std::ofstream myFile(“Net_Force_Table”)
   write the **Net_Force_Table.csv**, which is the result of Fruchterman-Reingold algorithm