A8

Due: See dropbox details
Submit: \.zip

Your Task

You are to build off of the graph implementation you built for a7 and add the following functionality to it.

```
dijkstra(int src);
primm(int src);
```

Details

For this assignment you can assume that the graph is *DIRECTED* and *WEIGHTED*. Your program needs to take this into account as Dijkstra's and Primm's do not function on undirected or unweighted graphs. The following files have been included for you to start with

- `graph.h` Defines the interface your class should implement
- `main.cpp` This is a minimum file which contains the driving logic you may use to test your program
- `makefile` This is the makefile which can be used to build your program
- `graph.cpp` This is the implementation file for your tree. This file is currently empty
- `edges.txt` This file contains the edge information needed to build your graph. It will be read by main.cpp to create an inial graph.
- `cmd.txt` This file will be read by main.cpp and determines what your program will do
- `output.txt` Contains the expected output for a correctly implemented program which runs the provided `cmd.txt` and `edges.txt`

cmd File Format

The cmd file for this assignment consists of two columns of numbers. The first column indications the function that should be called, and the second column indications the operand that that function may take. If the function takes no operand then the entry will be blank. The first column number to function mapping is as follows

```
1   setEdge         <param1> <param2>
2   printGraphData
3   dfs             <param1>
4   bfs             <param1>
5   dijkstra        <param1>
6   primm           <param1>
```

edges File Format

`edges.txt` follows the format provided below

- line 1: number of nodes in the graph provided by the file
- rest of the file: any number of source and destination node pairs with the weight for that edge

A sample file may look something like:

```
7
1 2 2
1 4 1
2 4 3
2 5 10
3 1 4
3 6 5
4 3 2
4 5 2
4 6 8
4 7 4
5 7 6
7 6 1
```

How I will test your program

To test your program I will use your `graph.h` and `graph.cpp` files and combine them with my own `main.cpp` file. My version of `main.cpp` will be similiar to the one that is provided with a few extra grading features and test cases added in.

Deliverables

Add `graph.h`, `bstgraph.cpp`, `main.cpp`, `makefile`, and `cmd.txt` to a zip file called `a8.zip` and submit that file to the dropbox. I will only accept submissions that meet these requirements.

Expections

- No use of any prebuilt graphing libraries
- Your code should be well formatted, points may be taken for sloppy code
- Your output should match the output provided in output.txt
- You must submit a zip file called `a8.zip`
- Your code should be fairly robust and be able to handle obvious edge cases
- You *MAY* use parts of the C++ Standard Template Library such as the stack and queue libraries.