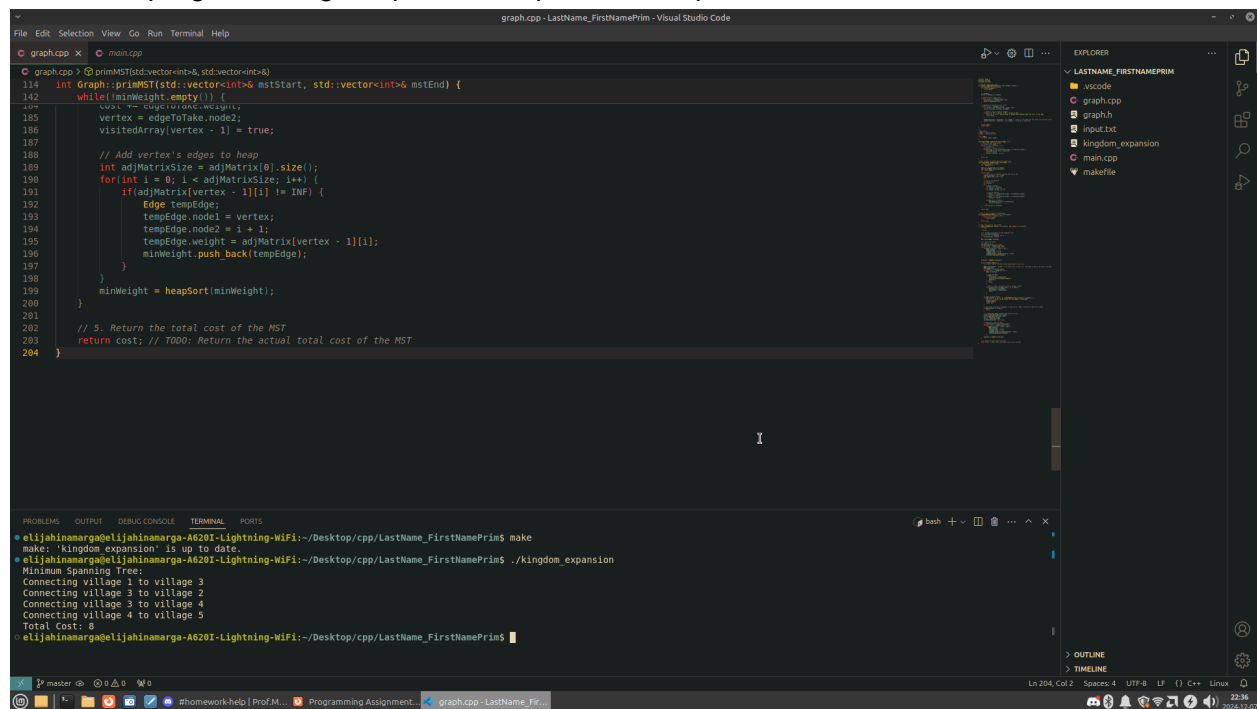


This is the program using the provided input.txt example



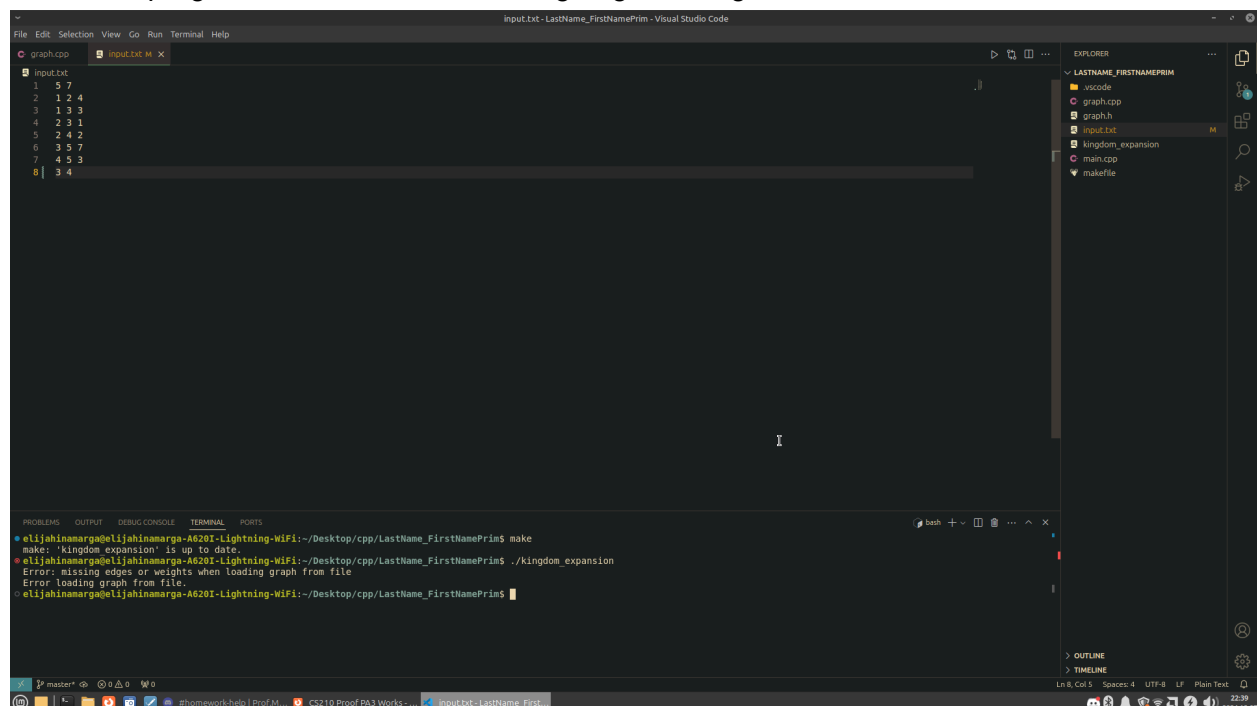
The screenshot shows a Visual Studio Code editor with a C++ file named `graph.cpp`. The code implements Prim's algorithm to find a Minimum Spanning Tree (MST) for a weighted undirected graph. The graph is represented by an adjacency matrix `adjMatrix` of size `n` by `n`, where `n` is the number of vertices. The algorithm starts with a single vertex (vertex 0) and iteratively adds the edge with the minimum weight that does not create a cycle or result in a vertex with a degree greater than 2. The total cost of the MST is calculated and returned.

```
graph.cpp:primMST(std::vector<int>& std::vector<int>&) {
114 int Graph::primMST(std::vector<int>& mstStart, std::vector<int>& mstEnd) {
115 while(!minWeight.empty()) {
116     vertex = edgeToTake.node2;
117     visitedArray[vertex - 1] = true;
118     // Add vertex's edges to heap
119     int adjMatrixSize = adjMatrix[0].size();
120     for(int i = 0; i < adjMatrixSize; i++) {
121         if(adjMatrix[vertex - 1][i] != INF) {
122             Edge tempEdge;
123             tempEdge.node1 = vertex;
124             tempEdge.node2 = i + 1;
125             tempEdge.weight = adjMatrix[vertex - 1][i];
126             minWeight.push_back(tempEdge);
127         }
128     }
129     minWeight = heapSort(minWeight);
130 }
131 // 5. Return the total cost of the MST
132 return cost; // TODO: Return the actual total cost of the MST
133 }
```

The terminal output shows the execution of the program. It reads the input from `input.txt` and prints the MST edges and the total cost.

```
elijahinamarga@elijahinamarga-A6201-Lightning-WiFi:~/Desktop/cpp/LastName_FirstNamePrim$ make
make: 'kingdom_expansion' is up to date.
elijahinamarga@elijahinamarga-A6201-Lightning-WiFi:~/Desktop/cpp/LastName_FirstNamePrim$ ./kingdom_expansion
Minimum Spanning Tree:
Connecting village 1 to village 3
Connecting village 3 to village 2
Connecting village 3 to village 4
Connecting village 4 to village 5
Total cost: 8
elijahinamarga@elijahinamarga-A6201-Lightning-WiFi:~/Desktop/cpp/LastName_FirstNamePrim$
```

This is the program when there is a missing edge or weight



The screenshot shows the `input.txt` file in the Visual Studio Code editor. The input consists of 8 lines of data, where the last line (line 8) contains a missing edge or weight, indicated by a vertical bar and a space.

```
input.txt:
1 5 7
2 1 2 4
3 1 3 3
4 2 3 1
5 2 4 2
6 3 5 7
7 4 5 3
8 | 3 4
```

The terminal output shows the execution of the program. It reads the input from `input.txt` and prints the MST edges and the total cost. However, it encounters an error: "Error: missing edges or weights when loading graph from file".

```
elijahinamarga@elijahinamarga-A6201-Lightning-WiFi:~/Desktop/cpp/LastName_FirstNamePrim$ make
make: 'kingdom_expansion' is up to date.
elijahinamarga@elijahinamarga-A6201-Lightning-WiFi:~/Desktop/cpp/LastName_FirstNamePrim$ ./kingdom_expansion
Error: missing edges or weights when loading graph from file
elijahinamarga@elijahinamarga-A6201-Lightning-WiFi:~/Desktop/cpp/LastName_FirstNamePrim$
```

This is the program when there is no MST for the graph

The screenshot shows a Visual Studio Code editor window with the following components:

- Editor:** A file named `input.txt` is open, containing a 9x3 grid of integers:

```
1 7 7
2 1 2 4
3 1 3 3
4 2 3 1
5 2 4 2
6 3 5 7
7 4 5 3
8 3 4 1
9 6 7 5
```
- EXPLORER:** The file explorer on the right shows a project structure for `LASTNAME_FIRSTNAMEPRIM`, including `.vscode`, `graph.cpp`, `graph.h`, `input.txt`, `kingdom_expansion`, `main.cpp`, and `makefile`.
- TERMINAL:** The terminal at the bottom shows the execution of a `make` command. The output indicates that the `kingdom_expansion` target is up to date, but an error occurred: `Error: no MST exists for this graph`. The `Minimum Spanning Tree:` section shows a `Total Cost: -1`.

The status bar at the bottom indicates the current file is `input.txt - LastName_First...` and the editor is in `UTF-8` encoding with `LF` line endings.