CS 3305/W02: Data Structures

Assignment 12 – Weighted Graphs Due 11/29/2022 @ 11:59 PM (100 points total)

GENERAL SUBMISSION REQUIREMENTS

Upload all files individually as specified, not as zip files, to Assignments in D2L. Do not email files.
Make sure your program compiles, runs and produces the correct output.
Ensure you have the correct file name(s), and author header, as specified in the Assignment.
Always use meaningful labels for prompts, inputs, and outputs.
Always use comments, indentation and whitespace as shown in examples.
Note: Never hard-code test data in the test program, unless explicitly stated in the assignment.
Always allow the user to enter the test data using a menu option.

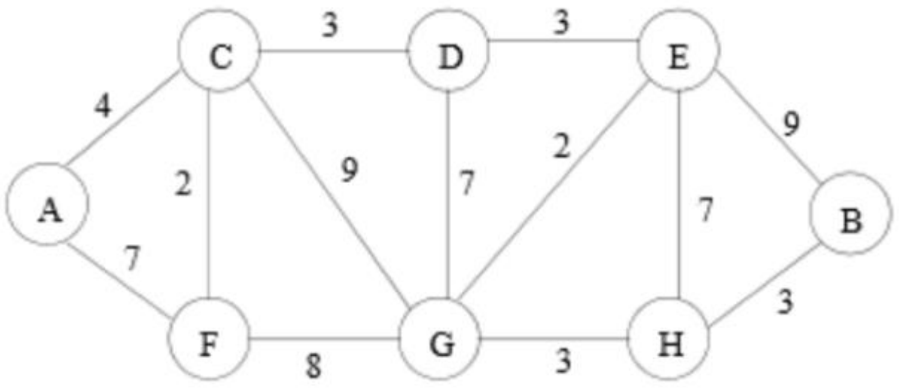
Assignment 12 – Part 1 Prim's REQUIRED (100 points) :

There are 3 Parts to this assignment. ONLY PART 1 IS REQUIRED * Part 2 & 3 are Extra Credit *

Part 1 is Programming Exercise 29.2, Prim's algorithm, from the Liang textbook, end of the Chapter Programming Exercises, re-printed below.

Implement Prim's algorithm using an adjacency matrix. The text implements Prim's algorithm using lists for adjacent edges. Implement the algorithm using an adjacency matrix for weighted graphs based on the graph provided below.

Be sure to list the order of the nodes visited, and, for Prim's always start at Node A. The distance traveled will be the answer we are looking for. The distance travelled should be the same for both Prim's and Kruskal's, but the difference between them will be the order of the nodes visited.



Do not forget to include author header in each submitted file as shown, no header, no points!

```
// Name:         <your name>
// Class:        CS 3305/ put your section number after the /
// Term:         Fall 2022
// Instructor:   Sharon Perry
// Assignment:   12-Part-1-Prims
```

Capture a **READABLE** screenshot(s) of your program output and paste into a word/pdf document. Readable means readable! Screenshots ***should not be an entire desktop*** – use some type of snipping tool. After your output screenshots, then copy and paste the source code for your program into the word/pdf doc. Save doc as a file named LastName-A12-Part-1-Prims.docx or .pdf. Last step is to upload word/pdf and .java files to D2L.

SUBMIT YOUR OWN CODE – Code copied from the internet will receive a score of zero.

MAKE SURE YOUR CODE HAS COMMENTS ! We are getting submissions without comments in the code. No comments = (-20) points *per Part of the assignment*.

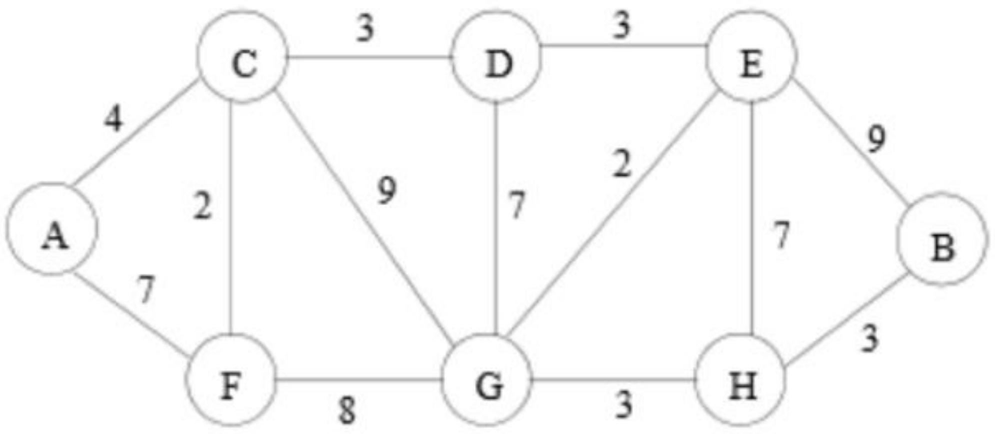
Assignment 12 – Extra Credit 1 Kruskal (25 points Extra Credit) :

Extra Credit 1 - Programming Exercise 29.1, Kruskals's algorithm, from the Liang textbook, end of the Chapter Programming Exercises, re-printed below.

The text introduced Prim's algorithm for finding a minimum spanning tree (MST).

Kruskal's algorithm is another well-known algorithm for finding a minimum spanning tree. The algorithm repeatedly finds a minimum-weight edge and adds it to the tree if it does not cause a cycle. The process ends when all vertices are in the tree.

Design and implement an algorithm for finding a minimum spanning tree (MST) using Kruskal's algorithm, using the graph provided below.



Be sure to list the order of the nodes visited. The distance traveled will be the answer we are looking for. The distance traveled should be the same for both Prim's and Kruskal's, but the difference will be the order of the nodes visited.

I know we tell you to study the textbook, but this video on Kruskal is so good we want to share it.
https://www.youtube.com/watch?v=HQbE9Xvc4ys

Do not forget to include author header in each submitted file as shown, no header, no points!

```
// Name:         <your name>
// Class:        CS 3305/ put your section number after the /
// Term:         Fall 2022
// Instructor:    Sharon Perry
// Assignment:    12-ExtraCredit-1-Kruskal
```

Capture a **READABLE** screenshot(s) of your program output and paste into a word/pdf document.

Readable means readable!   Screenshots ***should not be an entire desktop*** – use some type of snipping tool. After your output screenshots, then copy and paste the source code for your program into the word/pdf doc.   Save doc as a file named   LastName-A12-ExtraCredit-1-Kruskal.docx or .pdf. Last step is to upload word/pdf and .java files to D2L.

SUBMIT YOUR OWN CODE – Code copied from the internet will receive a score of zero.

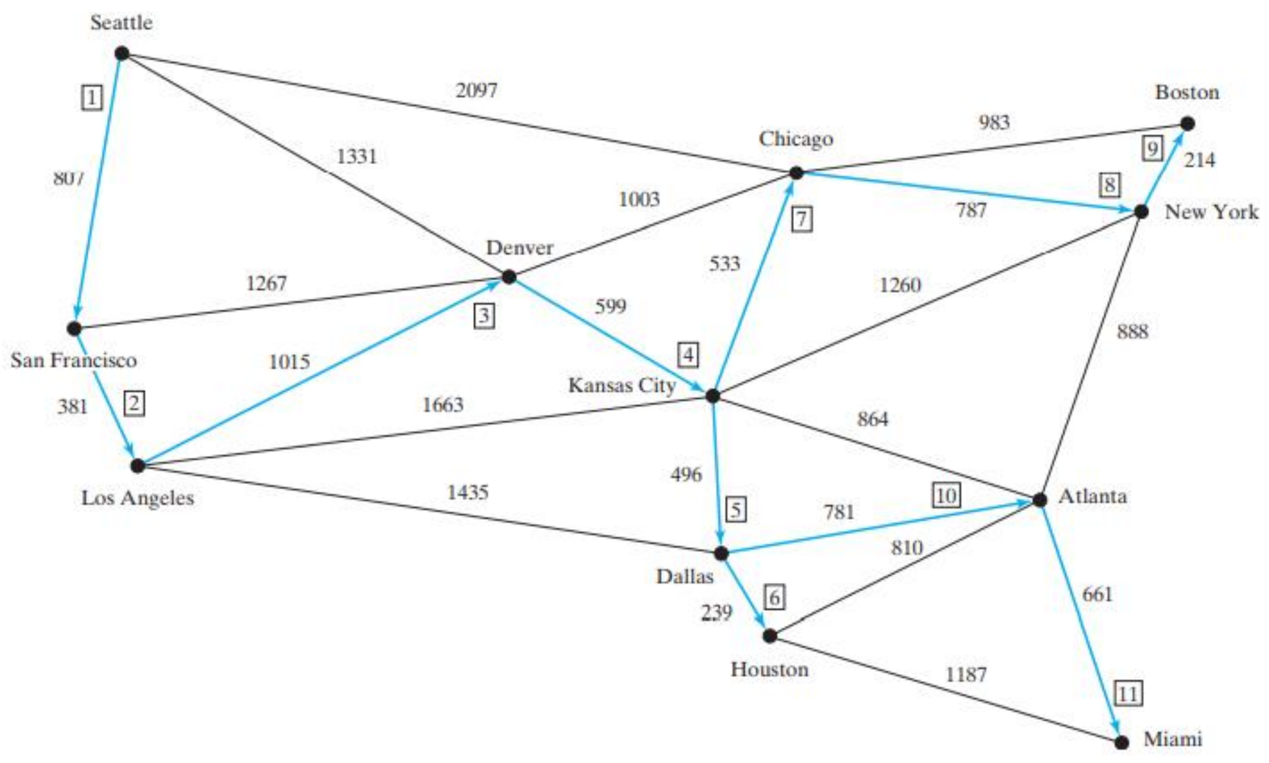
MAKE SURE YOUR CODE HAS COMMENTS ! We are getting submissions without comments in the code. No comments = (-20) points *per Part of the assignment*

Assignment 12 – Extra Credit 2   – Dijkstra   (25 points extra credit) :

Extra Credit 2 is Programming Exercise 29.3, Dijkstra's algorithm, from the Liang textbook, end of the Chapter Programming Exercises, re-printed below.

The text implements Dijkstra's algorithm using *lists* for adjacent edges. For this part of the extra credit, implement Dijkstra's algorithm using an adjacency matrix for weighted graphs, specifically, using the graph provided below.

Implement the algorithm using an adjacency matrix for weighted graphs, using the graph below.



Consider San Francisco as the source vertex. Therefore, your program needs to find the shortest distance from San Francisco:

to Seattle,
to Denver,

to Chicago,
to Los Angeles,
to Kansas City,
to Dallas,
to Houston,
to Atlanta,
to Boston,
to New York,
to Miami ….

In other words, we need the shortest path to EVERY vertex on the graph.

I know we tell you to study the textbook, but this video on Dijkstra's is so good we want to share it.
https://www.youtube.com/watch?v=5GT5hYzjNoo

Do not forget to include author header in each submitted file as shown, no header, no points!

```
// Name:         <your name>
// Class:        CS 3305/ put your section number after the /
// Term:         Fall 2022
// Instructor:    Sharon Perry
// Assignment:    12-ExtraCredit-2-Dijkstra
```

Capture a **READABLE** screenshot(s) of your program output and paste into a word/pdf document. Readable means readable! Screenshots ***should not be an entire desktop*** – use some type of snipping tool. After your output screenshots, then copy and paste the source code for your program into the word/pdf doc. Save doc as a file named LastName-A12-ExtraCredit-2-Dijkstra.docx or .pdf. Last step is to upload word/pdf and .java files to D2L.

SUBMIT YOUR OWN CODE – Code copied from the internet will receive a score of zero.

MAKE SURE YOUR CODE HAS COMMENTS ! We are getting submissions without comments in the code. No comments = (-20) points *per Part of the assignment*