# CS 211 PA #3

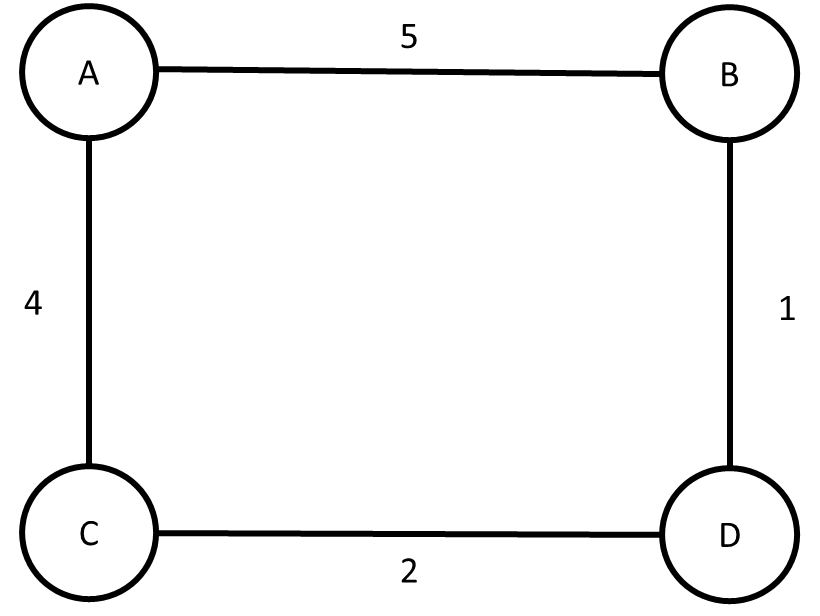
At UPS and other package delivery services, delivery trucks must navigate routes that change daily in a way that is most efficient. This assignment attempts to introduce you to the potential complexity of this task. In this assignment, you will be given a "map" of a city along with a set of houses that the delivery truck must visit. You then must use a Minimum Spanning Tree to compute the minimum amount of time required to deliver all packages.

## Maps File

The file format for the map is the same as PA #2: Each line is comprised of a source, skink, and weight. All edges are bi-directional. Below is the contents of map1.txt (included with this homework description):

|  |
| --- |
| A,B,5  A,C,4  C,D,2  B,D,1 |

This map corresponds to the following graph:



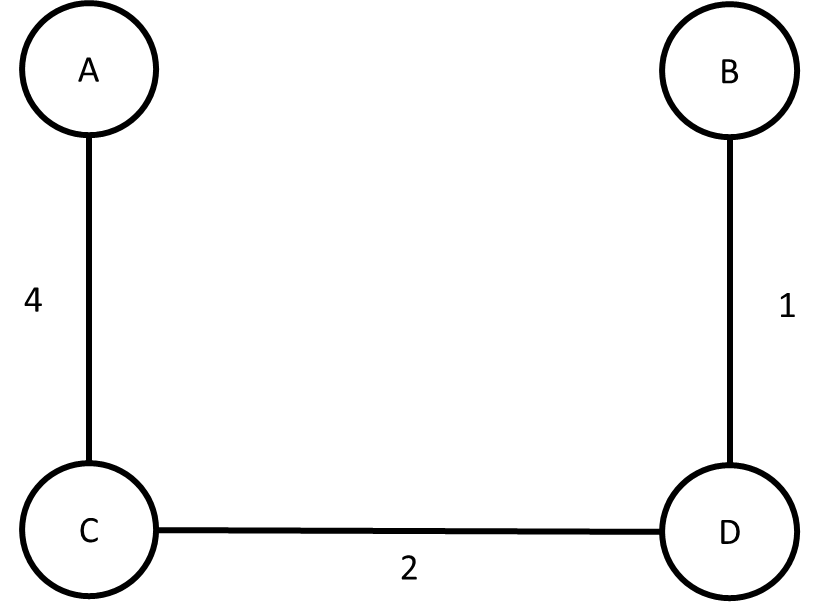
## Deliveries File

In a separate file contains a list of houses that will receive deliveries for the next day. There exists one house per line of the file. Below is the contents of deliveries1.txt:

|  |
| --- |
| A  B  C |

## Tier-1 Submissions

The basic submission requirements for this assignment is to construct an MST that connects all desired points using the least cost edges. Thus, starting with the first vertex in the destinations file (Vertex A), you would construct the following MST:

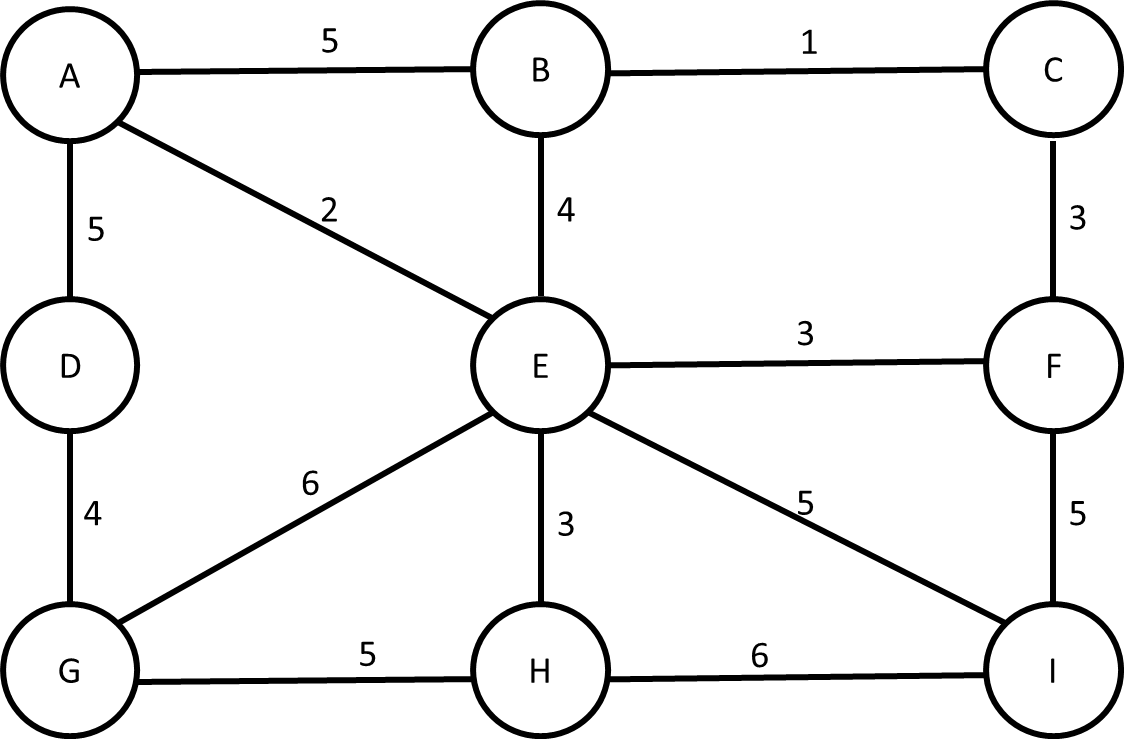


Your program must output the total cost of the route plus the accepted edges to the screen. Sample output:

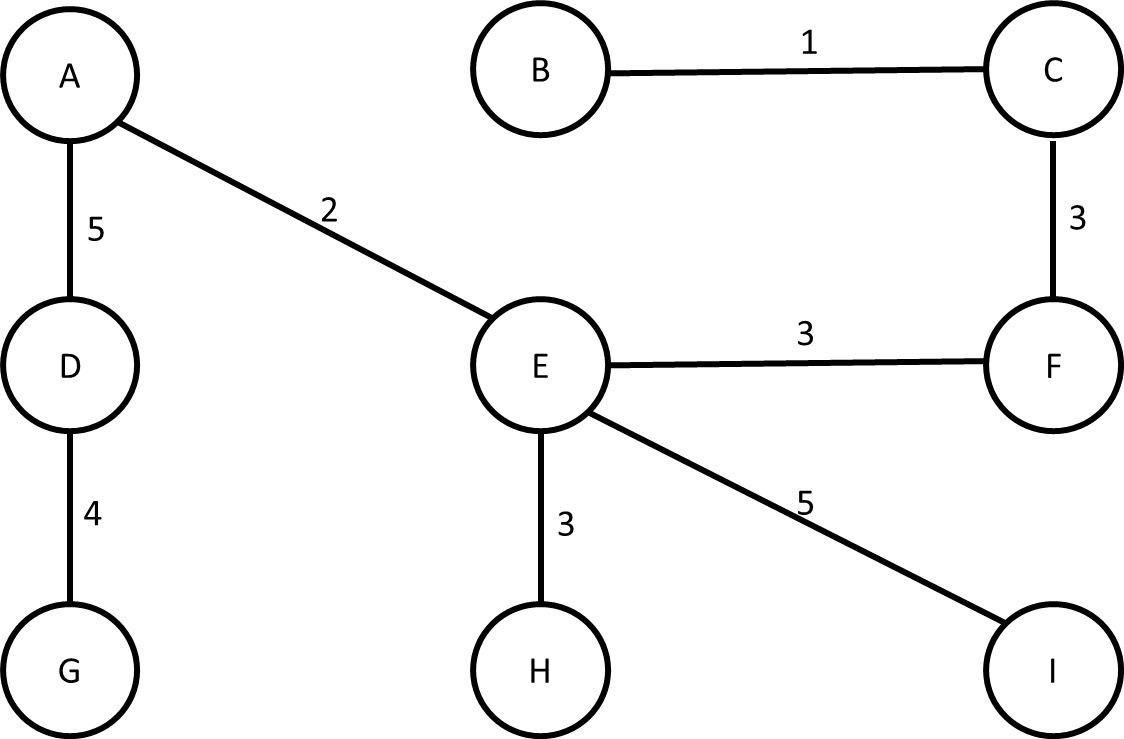
|  |
| --- |
| \*\*\*Route Planner\*\*\*  Enter maps file: **map1.txt**  Enter destinations file: **deliveries1.txt**  Total transit time: 7 minutes |

## Tier-2 Submissions

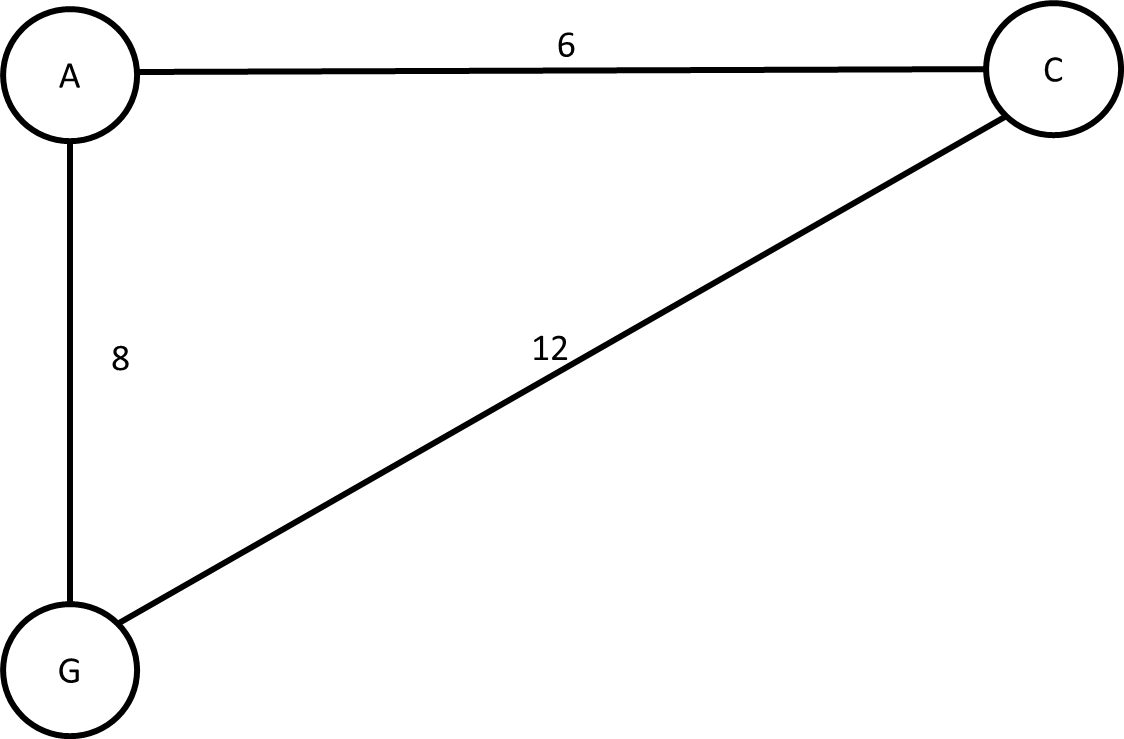
Note that an MST guarantees that all vertices in a graph will be connected using the least cost edges, but it does not guarantee that the connection between any two vertices are the most efficient. Given that our delivery routes may not deliver to all houses, we may inadvertently develop a suboptimal route. For example, consider the graph in maps2.txt:



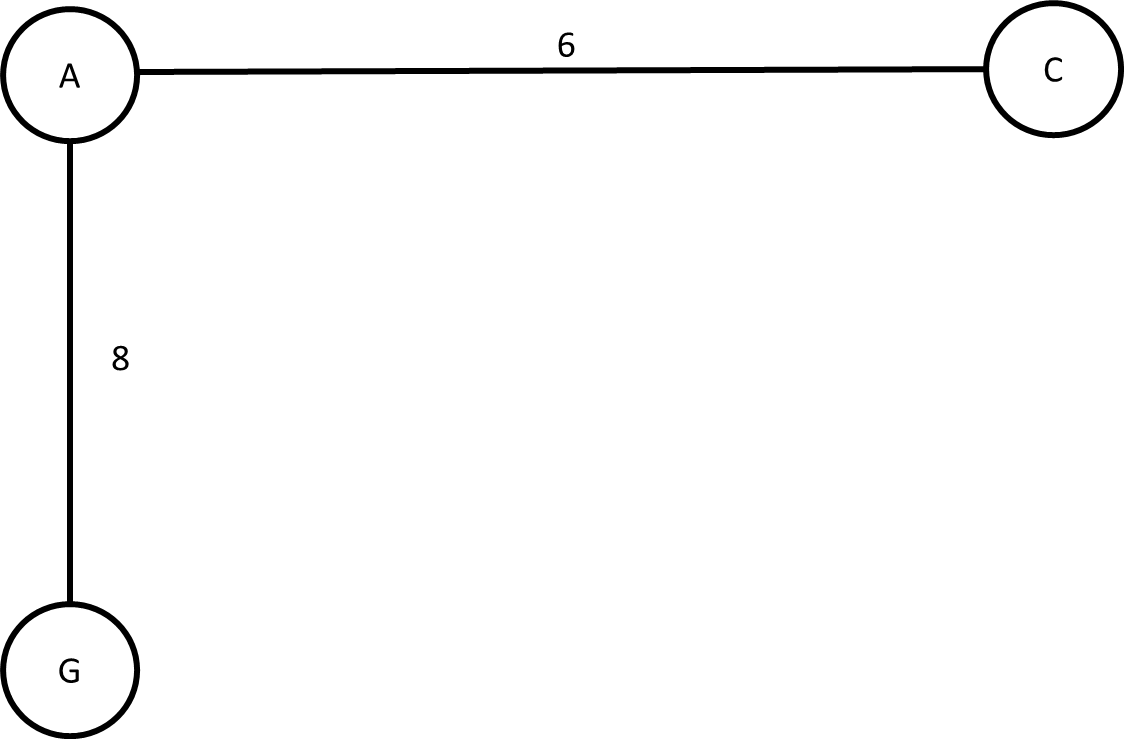
Assuming that we started again at Vertex A, we might compute the following MST:



However, what if our deliveries were only between A, C, and G? On the MST above, our route would be ADG then possibly GHEFC with a total travel time of 17 minutes. However, if we instead reduce the map to only the relevant vertices and connect them based on shortest path using Dijkstra's Algorithm, we may get a faster path. We begin by reducing the graph down to just vertices A, C, and G:



The edge weights for this reduced graph are derived from running Dijkstra's Algorithm at each vertex. Thus, it costs 8 minutes to get from A to G via the original route of AEG, 6 minutes to get to C via the original route ABC, and 12 minutes to get from C to G via original route CFEG. Building an MST on this tree yields:



Starting at Vertex A, our total route cost would be 14. Quite the improvement! Note that the output for Tier-2 submission will look exactly the same as a Tier-1 submission.

## Tier-3 Submissions

Tier-3 submissions add one additional requirement to the Tier-2 submission. Given the graph above, we know that the minimum time is 14 minutes, but we don't know the route required to make that time. A tier-3 submission will output the exact routes to take in order to achieve minimal time. Note that in the case above, either G->A->C or C->A->G are correct. Here's the updated output:

|  |
| --- |
| \*\*\*Route Planner\*\*\*  Enter maps file: **map2.txt**  Enter destinations file: **deliveries2.txt**  Total transit time: 14 minutes  Route:  G -> A  A -> C |

## Header Comment, and Formatting

1. Be sure to modify the file header comment at the top of your program to indicate your name, student ID, completion time, and the names of any individuals that you collaborated with on the assignment.
2. Remember to follow the basic coding style guide. For a list of basic rules, [see my website](http://adamcarter.com/teaching/cpts121/style) or examine my example files from previous assignments and labs.

## Reflection Essay

In addition to the programming tasks listed above, your submission must include an essay that reflects on your experiences with this homework. This essay must be at least 350 words long. Note that the focus of this paper should be on your reflection, ***not*** on structure (e.g. introductory paragraph, conclusion, etc.). The essay is graded on content (i.e. it shows deep though) rather than syntax (e.g. spelling) and structure. Below are some prompts that can be used to get you thinking. Feel free to use these or to make up your own.

* Describe a particular struggle that you overcame when working on this programming assignment.
* Conversely, describe an issue with your assignment that you were unable to resolve.
* Provide advice to a future student on how he or she might succeed on this assignment.
* Describe the most fun aspect of the assignment.
* Describe the most challenging aspect of the assignment.
* Describe the most difficult aspect of the assignment to understand.
* Provide any suggestions for improving the assignment in the future.

## Deliverables

The final version of your program must be uploaded through Canvas no later than midnight on Monday, March 11, 2019.

## Grading Criteria

This assignment is worth 100 points. Your assignment will be judged by the following criteria:

### Reflection (10pts)

* Your submission includes a reflection that meets the criteria set forth earlier in this document.

### Tier 1 (60pts)

* Your program meets tier 1 requirements

### Tier 2 (30pts)

* Your program meets tier 2 requirements

### Tier 3 (20 BONUS)

* Your program meets tier 3 requirements