Name: ______________________________ Andrew Id: ______________________

15-121 Fall 2023 Assessment 6

Up to 50 minutes. No calculators, no notes, no books, no computers. Show your work!

1. **Searching and Sorting**

   Consider the following list of integers:

   16, 8, 19, 6, 2, 1, 20, 13, 17

   (a) (3 points) Assuming that the array provided at the beginning of this problem receives a single pass from the loop in the Selection Sort algorithm (assume it is the first pass, with `i = 0`), what will be the new state of the array? Write your answer in the provided boxes below. Show your work. Answers without appropriate work will receive no credit.

   Final Answer for Part a:

   (b) (4 points) Assuming that the array provided at the beginning of this problem receives a single pass from the loop in the Bubble Sort algorithm, what will be the new state of the array? Write your answer in the provided boxes below. Show your work. Answers without appropriate work will receive no credit.

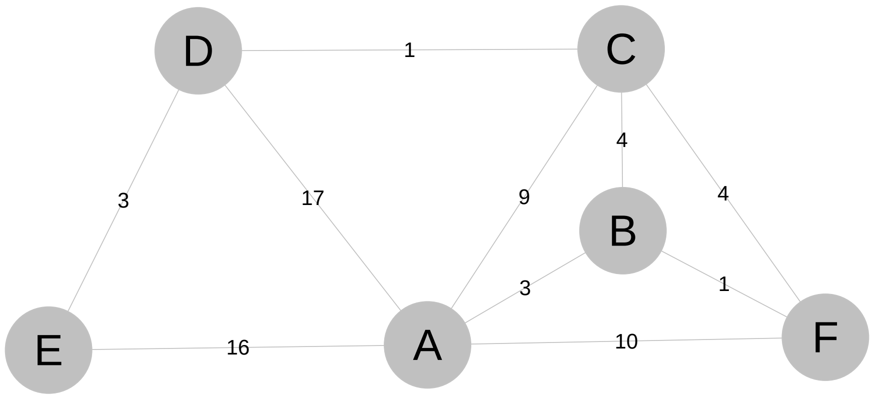
   Final Answer for Part b:

   (c) (5 points) Assuming that the array provided at the beginning of this problem is partitioned using the partition algorithm from Quick Sort (use 16 as your pivot), what will be the new state of the array? Write your answer in the provided boxes below. Show your work. Answers without appropriate work will receive no credit.

   Final Answer for Part c:

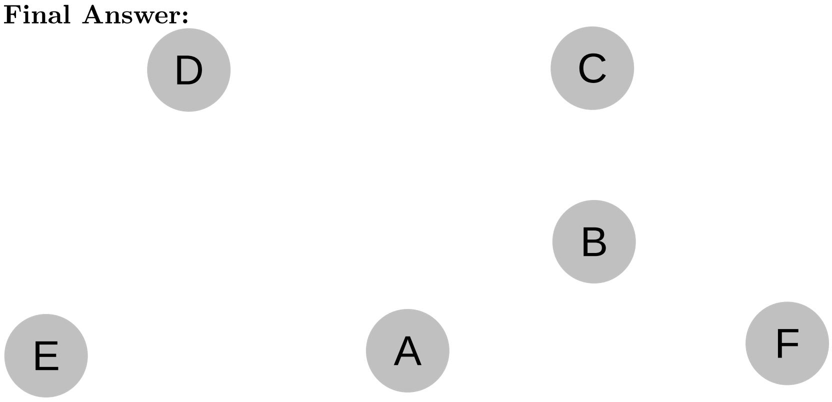
2. **Graphs**

Consider the following weighted graph:



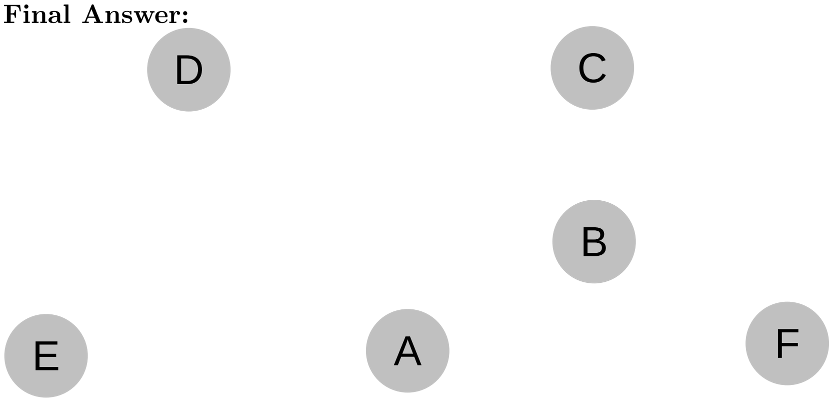
(a) (6 points) Find a minimum spanning tree for this graph using Kruskal's algorithm. Be sure to show your work in such a way that it is clear you understand how the algorithm works (for example, by explictly stating the order in which edges are added to the MST). Answers without appropriate work will receive no credit. Draw your final MST onto the provided graph below.

Final Answer:

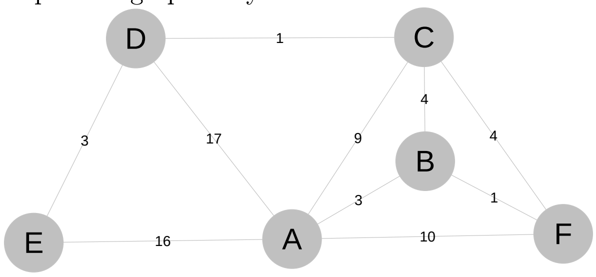


(b) (6 points) Find a minimum spanning tree for this graph using Prims's algorithm starting from E. Be sure to show your work in such a way that it is clear you understand how the algorithm works (for example, by explictly stating the order in which edges are added to the MST). Answers without appropriate work will receive no credit. Draw your final MST onto the provided graph below.

Final Answer:



Reprint of graph for your convenience:



(c) (10 points) Using Dijkstra's algorithm, find the shortest paths from E to every node in the graph. Be sure to show your work. Answers without appropriate work will receive no credit.

Write your final answer (your done / visited table) in the table below. Do *not* write your unvisited / working table in the table below.

Final Answer:

Node	**Cost**	**Previous**
E	0	

3. Graph Code

Recall the following code, written in class, for Prim's algorithm:

```java
public Graph MSTPrims() {
    HashSet<Vertex> verticesInMST = new HashSet<Vertex>();
    ArrayList<Edge> sortedEdges = new ArrayList<Edge>();
    HashSet<Edge> finalEdges = new HashSet<Edge>();

    Vertex starter = null;
    for (String s : this.vertices.keySet()) {
        starter = this.vertices.get(s);
        break;
    }
    verticesInMST.add(starter);


    sortedEdges.addAll(starter.getEdges());
    Collections.sort(sortedEdges);

    while (verticesInMST.size() < this.vertices.size()) {
        Edge e = sortedEdges.remove(0);

        if (!verticesInMST.contains(e.getDst())) {
            verticesInMST.add(e.getDst());

            sortedEdges.addAll(e.getDst().getEdges());
            Collections.sort(sortedEdges);

            finalEdges.add(e);
        }
    }

    Graph MST = new Graph();
    for (Edge e : finalEdges) {
        MST.addVertex(e.getSrc().getName());
        MST.addVertex(e.getDst().getName());
        MST.addEdge(e.getSrc().getName(), e.getDst().getName(), e.getWeight());
        MST.addEdge(e.getDst().getName(), e.getSrc().getName(), e.getWeight());
    }
    return MST;
}
```

Also consider the following definition of a min-heap:

`public class MinHeap<DataType extends Comparable<DataType>>`

A min-heap implementation.

`public void add(DataType item)`

Add an item to the heap. Ensures that the heap property is maintained.

- **Parameters:** `item` — The item to add to the heap.

`public DataType removeMin()`

Get the minimum item in the heap, remove it from the heap, and return it. Ensures that the heap property is maintained.

- **Returns:** The minimum item from the heap.

(a) (4 points) Assuming that there is a working implementation of `MinHeap` (meaning that you don't need to write it) modify the `MSTPrims` method to be use it to be more efficient. This means that you will need to analyze the Prim's algorithm code, determine how a heap could improve it, and then modify the Prim's code appropriately to use a heap.

Write your answer directly onto the code of the previous page. If you need more space, then write something like "See code block A" and then provide code on this page that is labeled as "Code block A".

(b) (2 points) Discuss how using a heap impacts the efficiency of the algorithm. Be specific, citing the efficiency of certain piece of code before using a heap, and the efficiency after using a heap. You do not need to discuss the overall efficiency of Prim's algorithm. Instead, just discuss the efficiency improvements of the pieces of it which you modified.