1. (4 points) Consider the following function from FlowEdge.java. This function is used to allocate flow to an edge in a residual graph. Fill in the blanks with the appropriate type of edge.

```java
public void addResidualFlowTo(int vertex, double delta) {

    // allocate flow to a _BACKWARDS______ edge
    if (vertex == v) flow -= delta;

    // allocate flow to a _FORWARD_______ edge
    else if (vertex == w) flow += delta;

    else throw new IllegalArgumentException("Illegal endpoint");

    if (Double.isNaN(delta))
        throw new IllegalArgumentException("Change in flow = NaN");
    if (!(flow >= 0.0))
        throw new IllegalArgumentException("Flow is negative");
    if (!(flow <= capacity))
        throw new IllegalArgumentException("Flow exceeds capacity");
}
```

2. (2 points) What is the runtime of Eager Prim’s?

\( \Theta(elgv) \)

3. (2 points) What is the runtime of Dijkstra’s algorithm?

\( \Theta(elgv) \)
4. (8 points) Give the pseudocode for Kruskal’s algorithm.

Add all edges in the graph to a PQ. Remove min edges from the PQ and add them to the MST so long as they do not create a cycle in the MST.

5. (6 points) Explain how union find data structures can be used to implement cycle detection.

UF efficiently determines connectedness of two points. When adding edges one at a time, you can determine that if the endpoints of an edge to be added are already connected in the graph, then adding this new edge would cause the endpoints to be biconnected, creating a cycle.

6. (10 points) Use Edmonds Karp to find the max flow of the following graph. Consider verticies to be seen in alphabetical order. In addition to stating the max flow, list the augmenting paths that you find in the order you use them, and the flow that you push along each.

Max flow: 17
s A t: 2
s B E t: 5
s C D t: 7
s B C D t: 3
7. (10 points) Use Dijkstra’s algorithm to determine the shortest path from vertex A to vertex G in the following graph. In addition to stating the shortest path from A to G, state the order in which you visit the vertices of the graph in finding this shortest path.

Shortest path: A D F G of weight 20
Visited order: A B C D E F G

8. (8 points) Consider computer file storage. Folders can contain both files and other folders. Assume that there exists one root folder. If you want to write a program to print, for each folder on the computer, the count all of the files contained within that folder (including files contained within all of its subfolders, files within all of their subfolders, etc), what approach would you take? Describe your approach using vocabulary discussed in class wherever possible.

Treat the hierarchy of folders like a tree. Files and folders are nodes in the graph. Perform a DFS post-order traversal starting at the root. Post-order to count all the files stored in subfolders, then sum them together and print out the total for the current folder after you have recursed.