1. **[5 points]** Does the following represent a flow in a flow network? If not, explain why not.

   ![Flow Network 1](image1)

2. **[10 points]** Consider the following network and flow.

   ![Flow Network 2](image2)

   (a) Draw the residual network for this flow.
   (b) Given an augmenting path for this flow.
   (c) Increase the flow by the maximum possible value along the augmenting path and draw the resulting flow network and flow.

3. **[10 points]** Prove that the following flow is maximum.

   ![Flow Network 3](image3)

4. **[15 points]** Consider the following flow network. (The values are edge capacities.)

   ![Flow Network 4](image4)
(a) What is the value of the capacity of the cut \((S, T)\) where \(S = \{s, v_5, v_6\}\) and \(T = \{v_1, v_2, v_3, v_4, t\}\)?

(b) Prove that the cut \((S, T)\) where \(S = \{s, v_5, v_6\}\) and \(T = \{v_1, v_2, v_3, v_4, t\}\) is a minimum cut. (Hint: Use the max-flow min-cut theorem.)

5. [10 points] Given the following maximum flow in a flow network, give the minimum cut whose value equals the maximum flow.

![Flow Network Diagram](attachment:flow-network.png)

6. [10 points] Suppose you are given a directed graph \(G = (V, E)\), such that between any two nodes \(u, v \in V\), there is at most one directed edge. Let \(s, t \in V\) be two nodes in \(V\). Design an algorithm so that it can compute the least number of edges we need to remove from \(E\) so that \(t\) is not reachable from \(s\). You do not need to write a pseudo-code as long as you can describe your algorithm clearly. Provide a time complexity analysis of your algorithm. (Slower algorithms receive fewer points.)

7. [10 points] Given an undirected graph \(G = (V, E)\), describe an algorithm to check whether this input graph is a bipartite graph or not. You do not need to write a pseudo-code as long as you can describe your algorithm clearly. Provide a time complexity analysis of your algorithm. (Slower algorithms receive fewer points.)