Mid-Term Examination
Mathematical Programming with Applications to Economics
Total Score: 50

1. Show that if every cycle of an undirected graph has even number of edges then it is a bipartite graph. (**5 marks**)

   Answer: Suppose we have an undirected graph in which every cycle has even number of edges. We say that two vertices i and j are at even (odd) distance if there is a path from i to j which has even (odd) number of edges. Note that if there is more than one path from i to j, then all paths have either odd number of edges or even number of edges - otherwise, the corresponding cycle will have odd number of edges, a contradiction.

   Now, we search for all vertices in this graph and label them b or l. Start from any vertex i, and label it b. All vertices which are at odd distance from i get label l and all vertices which are at even distance from i get label b. Then, look for a vertex which is unlabeled, and repeat the procedure. It is clear that letting B equal all vertices with label b and letting L equal all vertices with label l defines a bipartite graph.

2. Suppose $G = (N, E, w)$ is a strongly connected digraph which has no cycles of negative length. Fix a node $i \in N$. Let p be defined as $p(i) = 0$ and $p(j) = -s(j, i)$ for all $j \in N \setminus \{i\}$, where $s(j, i)$ is the shortest path length from j to i in G.

   - Show that p is a potential of G. (**5 marks**)

     Answer: Consider any edge $(u, v) \in E$. Consider the shortest path from v to i. There are two cases to consider:

     Case 1: The shortest path from v to i does not include vertex u. In that case, the direct edge from u to v followed by the shortest path from v to i defines a path from u to i. By the definition of shortest path length, $s(u, i) \leq w(u, v) + s(v, i)$.

     Case 2: The shortest path from v to i includes vertex u. In that case, $s(v, i) = s(v, u) + s(u, i)$. Hence, $s(v, i) + w(u, v) = w(u, v) + s(v, u) + s(u, i) \geq s(u, i)$, where the inequality follows from the fact that the cycle involving direct edge from u to v and the shortest path from v to u.

     So, in both cases we have $s(u, i) \leq w(u, v) + s(v, i)$, which is equivalent to saying $-p(u) \leq w(u, v) - p(v)$ or $p(v) - p(u) \leq w(u, v)$. Hence, p is a potential of G.

   - Let q be any other potential of G such that $q(i) = 0$. Show that $p(j) \leq q(j)$ for all $j \in N$, where potential p is as defined above. (**5 marks**)

Answer: Consider any vertex $j \in N$. Now, let $P = (j, j^1, \ldots, j^k, i)$ be the shortest path from j to i. We know that $s(j,i) = w(j, j^1) + \ldots + w(j^k, i) \geq q(i) - q(j) = -q(j)$, where the inequality followed because q is a potential and the last equality followed since $q(i) = 0$. Hence, $q(j) \geq -s(j,i) = p(j)$.

3. Construct the residual graph for the flow shown in the flow graph in Figure 1. Conclude from this if the flow is a maximum flow or not. (**5 marks**)

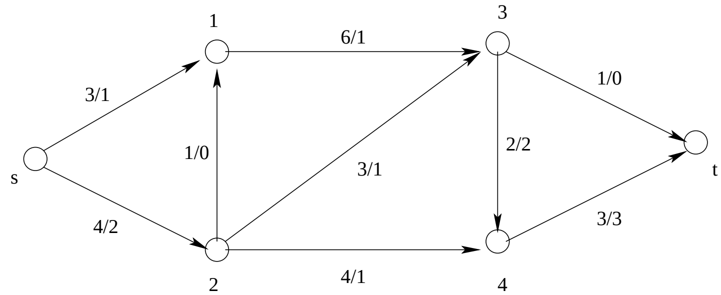

Figure 1: A flow graph

Answer: The residual graph for flow in Figure 1 is shown in Figure 2. There are may paths from s to t in this residual graph. One of them being: $(s, 2, 3, t)$, where a flow of 1 can be increased. So, the given flow is not a maximum flow.

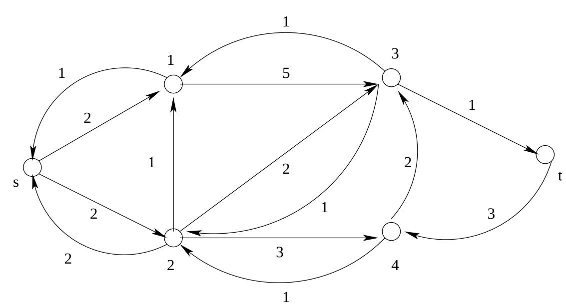

Figure 2: The residual graph for flow in Figure 1

4. Let $G = (N, E, w)$ be a weighted undirected graph. For every spanning tree $T = (N, E')$ of G, define a **bottleneck edge** of T as an edge which has the maximum weight over all edges in E'. A **minimum weight bottleneck spanning tree** is a spanning tree such that the weight of its bottleneck edge is the minimum over all spanning trees.

- Show that a minimum cost (weight) spanning tree is also a minimum weight bottleneck spanning tree. (**10 marks**)

Answer: Let $T = (N, E')$ be the minimum cost spanning tree of graph $G = (N, E, w)$. Let $\{i, j\}$ be the maximum weight edge (bottleneck edge) of T. Fix node i and consider all nodes k in $N \setminus \{i\}$ such that the unique path from i to k in T includes edge $\{i, j\}$. Call this set of nodes S. Note that $i \notin S$ and $j \in S$. So, S and $N \setminus S$ are non-empty sets. Hence, we can identify a cut $(S, N \setminus S)$ such that $\{i, j\}$ is the unique edge in E' which crosses this cut. Let $\{k, l\}$ be the edge which crosses this cut and belongs to minimum weight bottleneck spanning tree. Since T is the minimum cost spanning tree of G we have $w(\{i, j\}) \leq w(\{k, l\})$. If $\{u, v\}$ is the bottleneck edge of the minimum weight bottleneck spanning tree then $w(\{i, j\}) \leq w(\{k, l\}) \leq w(\{u, v\})$. Hence, T must also be a minimum weight bottleneck spanning tree.

- Use this to find a minimum weight bottleneck spanning tree for the graph in Figure 3. (**5 marks**)

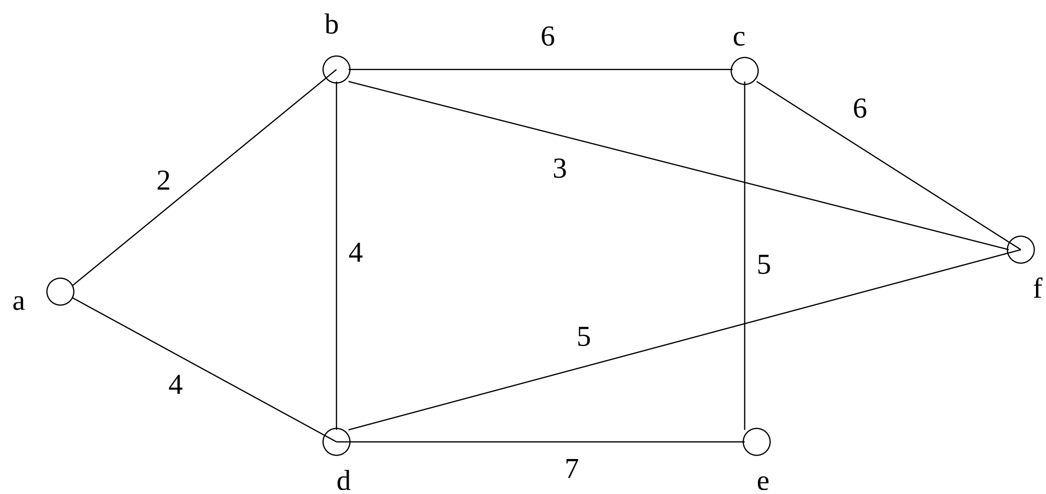

Figure 3: Finding minimum weight bottleneck spanning tree

Answer: By the previous question, we know that we only need to find an MCST. This can be found by the greedy algorithm discussed in the class. Verify that an MCST has edges: $\{b, a\}, \{b, c\}, \{b, d\}, \{b, f\}, \{c, e\}$.

5. For the bipartite graph in Figure 4, a matching M is shown with dark edges. For this matching M, either find an M-augmenting path or conclude that it is a maximum matching. What is the size of the minimum vertex cover of this bipartite graph? (**5 + 2 marks**)

   Answer: The required directed graph is show in Figure 5. Here, we can see that there is a path starting from a unmatched vertex on one side and ending at an unmatched vertex on the other side: $(d, 2, b, 3)$. This is an augmenting path for the matching in

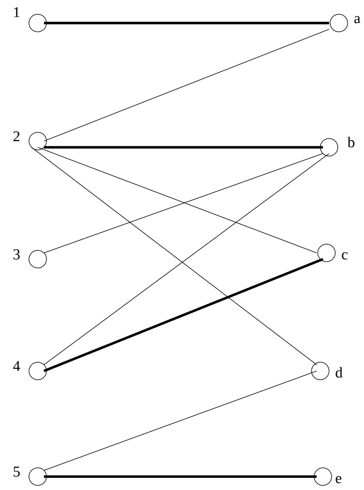

Figure 4: A bipartite graph

Figure 4. The new matching is thus: $\{1, a\}, \{2, d\}, \{3, b\}, \{4, c\}, \{5, e\}$. This has five edges, and hence a maximum matching (no matching can have more than five edges in this bipartite graph since each side has exactly five vertices). By the maximum size matching and minimum size vertex cover theorem, the size of the minimum vertex cover for this bipartite graph is 5.

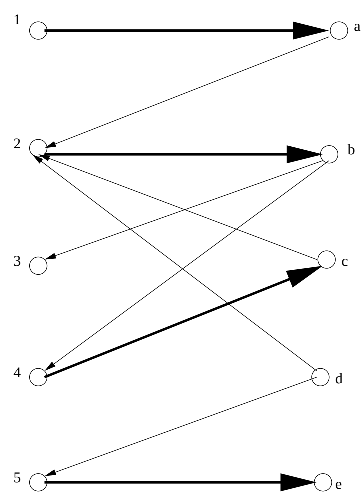

Figure 5: Augmenting path for the matching in bipartite graph of Figure 4

6. Consider the digraph in Figure 6. We need to find the minimum number of edges that need to be removed such that there are no paths from s to t in this digraph.

   - Construct the underlying flow graph for the digraph in Figure 6 to find the edge-disjoint paths. (**2 marks**)

- Compute the maximum flow of this flow graph. You may or may not use the algorithm to compute the maximum flow, but if you write some flow to be a maximum flow argue why it is so. (**3 marks**)
- Now, find the minimum number of edges that need to be removed such that there are no paths from s to t in the digraph in Figure 6. (**3 marks**)

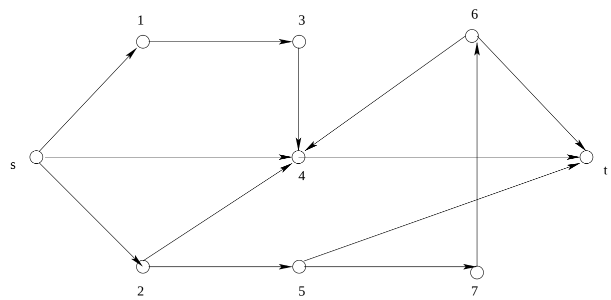

Figure 6: Edge disjoint path

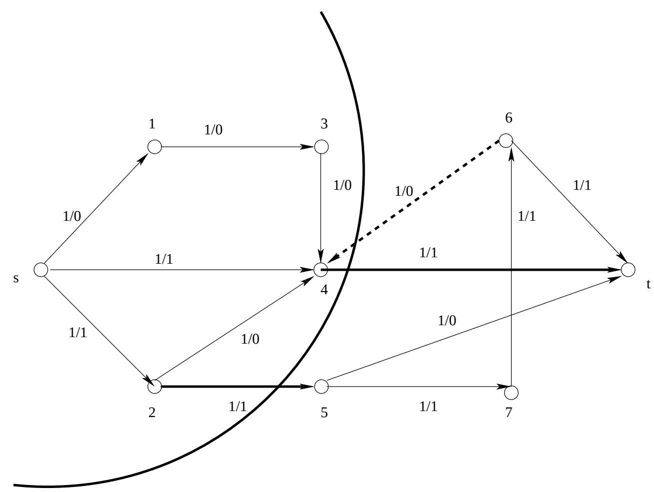

Figure 7: Edge disjoint path

Answer: The underlying flow graph has same set of vertices and edges with edge capacity of 1. A feasible flow of this graph is shown in Figure 7. Also shown is an (s,t)-cut of this flow graph: $(\{s,1,2,3,4\},\{5,6,7,t\})$. The optimality of this feasible flow can be seen from this cut in two ways: (a) the capacity of this cut is 2 and the value of this feasible flow is also two (so max-flow min-cut theorem says this is the maximum flow) and (b) this is a saturated cut for this flow since edge $\{4,t\}$ and $\{2,5\}$ have maximum possible flow but edge $\{6,4\}$ has zero flow.

The maximum flow in Figure 7 shows two disjoint paths from s to t: $(s,4,t)$ and $(s,2,5,7,6,t)$. Hence, there is a maximum of two disjoint paths from s to t. By

Menger's theorem, we need to remove a minimum of two edges such that there are no paths from s to t.