1. [5pts] (Primal-Dual Simplex) Solve the shortest st-path problem using the primal-dual simplex algorithm.
2. [5pts] (Dijkstra's Algorithm) Use Dijkstra's Algorithm to compute the distances from $s$ to all other vertices in the graph above.
Labels are distances from $s$. 

Start with zero flow.

Labelling Algorithm gives following paths:
(May be slightly different)

Augmenting Path!

By this cut of capacity $2t+1=3$, our flow has value 3 is optimal!
4. [5pts] (Floyd-Warshall Algorithm) Use the Floyd-Warshall Algorithm to compute shortest distances among all pairs of vertices in the graph given by the following adjacency matrix. (The entry $a_{i,j}$ stores the length of the arc $(i,j)$.)

\[
\begin{bmatrix}
\infty & 1 & 2 & \infty & \infty \\
\infty & \infty & \infty & 4 & \infty \\
6 & 1 & \infty & \infty & 3 \\
5 & 3 & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & 3 \\
\end{bmatrix}
\]

We perform the triangle updates on the five vertices in order (top-to-bottom/left-to-right). New values are in bold.

1. $v_1$

\[
\begin{bmatrix}
\infty & 1 & 2 & \infty & \infty \\
\infty & \infty & \infty & 4 & \infty \\
6 & 1 & 8 & \infty & 3 \\
5 & 3 & 7 & \infty & \infty \\
\infty & \infty & \infty & 3 & \infty \\
\end{bmatrix}
\]

2. $v_2$

\[
\begin{bmatrix}
\infty & 1 & 2 & 5 & \infty \\
\infty & \infty & \infty & 4 & \infty \\
6 & 1 & 8 & 5 & 3 \\
5 & 3 & 7 & 8 & \infty \\
\infty & \infty & \infty & 3 & \infty \\
\end{bmatrix}
\]

3. $v_3$

\[
\begin{bmatrix}
8 & 1 & 2 & 5 & 5 \\
\infty & \infty & \infty & 4 & \infty \\
6 & 1 & 8 & 5 & 3 \\
5 & 3 & 7 & 8 & 10 \\
\infty & \infty & \infty & 3 & \infty \\
\end{bmatrix}
\]

4. $v_4$

\[
\begin{bmatrix}
8 & 1 & 2 & 5 & 5 \\
9 & 7 & 11 & 4 & 14 \\
6 & 1 & 8 & 5 & 3 \\
5 & 3 & 7 & 8 & 10 \\
8 & 6 & 10 & 3 & 13 \\
\end{bmatrix}
\]

5. $v_5$

\[
\begin{bmatrix}
8 & 1 & 2 & 5 & 5 \\
9 & 7 & 11 & 4 & 14 \\
6 & 1 & 8 & 5 & 3 \\
5 & 3 & 7 & 8 & 10 \\
8 & 6 & 10 & 3 & 13 \\
\end{bmatrix}
\]
5. [5pts] Consider the following linear program.

\[ \begin{align*}
\text{max} \quad & x_1 + x_2 \\
\text{subject to} \quad & -\frac{8}{3}x_1 + x_2 \geq -\frac{8}{3} \\
& x_1 - x_2 \geq -\frac{1}{2} \\
& x_1, \quad x_2 \geq 0
\end{align*} \]

Solve the linear problem graphically, then also solve the problem graphically when \( x_1 \) and \( x_2 \) are constrained to be integers, demonstrating a gap between the real and integer solutions.

The constraints are bounds at the following lines:

\[ \begin{align*}
x_2 &= \frac{8}{3}x_1 - \frac{8}{3} \quad \text{(line A)} \\
x_2 &= x_1 + \frac{1}{2} \quad \text{(line B)}
\end{align*} \]

Plot:

Optimal Real Point:
\[ x_1 = \frac{19}{10}, \quad x_2 = \frac{24}{10} \quad \text{with value } \frac{43}{10} \]

Optimal Integer Point:
\[ x_1 = 2, \quad x_2 = 1 \quad \text{with value } 2 \]

\[ 2 < \frac{43}{10} \]
6. [10 pts] (Matchings and Vertex Covers) Let G be a graph with edges spanning two sets of vertices, X and Y. A matching is a set M of edges, where \( M = \{x_iy_i : 1 \leq i \leq k \} \) for some k, \( x_i \in X \), and \( y_i \in Y \), with \( x_iy_i \in E(G) \). A vertex cover is a set \( Q \subset V(G) \) such that all edges have at least one endpoint in Q. Use Max-Flow/Min-Cut to prove that the maximum size of a matching in a bipartite graph G is equal to the minimum size of a vertex cover. (Hint: Add vertices s and t to G, direct the edges, and show that the max st-flow and min st-cut problems are equivalent to the max matching and min vertex cover problems.)

Proof. Given a bipartite graph G with bipartition X \( \cup Y \), we will build a network N whose flows correspond to matchings of G and whose minimum cuts correspond to minimum vertex covers in G.

Let N have vertex set \( V(N) = \{s, t\} \cup X \cup Y \). For each \( x \in X \), let \( sx \) be an edge of capacity 1. For each \( y \in Y \), let \( yt \) be an edge of capacity 1. For each edge \( xy \in E(G) \), let \( xy \) be an edge of N with capacity \( |X| + |Y| \). Since the capacities are integers, the Ford-Fulkerson algorithm guarantees that maximum flows will have integer values on the edges.

Given a feasible integer flow f in N, let \( M_f = \{xy : x \in X, y \in Y, f(xy) = 1\} \). Since each \( x \in X \) has a maximum incoming flow of 1, there is at most one edge \( xy \in M_f \). Since each \( y \in Y \) has a maximum outgoing flow of 1, there is at most one edge \( xy \in M_f \). Thus, \( M_f \) is a matching and observe that \( |M_f| \) is equal to the value of f.

Given a matching M in G, let f be a flow defined as

1. \( f(sx) = 1 \) if and only if \( x \) is saturated by M,
2. \( f(yt) = 1 \) if and only if \( y \) is saturated by M, and
3. \( f(xy) = 1 \) if and only if \( xy \in M \),

where \( x \in X \) and \( y \in Y \). Observe that since M is a matching, f is a feasible flow in N and f has value equal to \(|M|\).

Let \([W, \overline{W}]\) be a minimum st-cut in N. Since assigning \( W = \{s\} \) or \( \overline{W} = \{t\} \) presents a cut of capacity \(|X| \) or \(|Y|\), a minimum st-cut \([W, \overline{W}]\) never contains an edge from X to Y, since their capacities are strictly larger. Thus, let \( Q = (\overline{W} \cap X) \cup (W \cap Y) \). We claim that Q is a vertex cover of size equal to the capacity of \([W, \overline{W}]\). Observe that no edges span \( \overline{W} \cap X \) and \( W \cap Y \) or else the capacity of the cut is too large (by earlier argument). Thus, every edge \( xy \in E(G) \) has at least one endpoint in Q, so Q is a vertex cover. Also, since \( sx \in [W, \overline{W}] \) for all \( x \in Q \) and \( yt \in [W, \overline{W}] \) for all \( y \in Q \), the capacity of \( W \) is equal to the size of Q.

For any vertex cut Q, let \( W = \{s\} \cup (X \setminus Q) \cup (Y \setminus Q) \). We claim that the capacity of W is equal to the size of Q: if \( x \in Q \cap X \), then \( sx \in [W, \overline{W}] \); if \( y \in Q \cap Y \), then \( yt \in [W, \overline{W}] \). Since Q is a vertex cover, no edges from X to Y are in \([W, \overline{W}]\), so hence the capacity of this st-cut is equal to \(|Q|\).

Since the size of a maximum matching equals the value of a maximum flow, the value of a maximum flow equals the capacity of a minimum cut, and the capacity of a minimum cut is the size of some vertex cover, we have the maximum matching is bounded below by the minimum size of a vertex cover. Since a minimum vertex cover has size equal to the capacity of an st-cut, the capacity of an st-cut is at least the value of a maximum flow, and the value of a maximum flow is the size of a maximum matching, we have the minimum vertex cover is bounded below by the maximum matching. Thus, the size of a maximum matching is equal to the size of a minimum vertex cover. \( \square \)