1. We are given a general digraph $G$ with edge capacities $u_e \geq 0$. Also, for every node $v \in V$, we have a real number $b_v$ (not necessarily nonnegative).

i. Give an efficient algorithm to solve the following general version of the transshipment problem discussed in class. Find a set of real numbers $x_e$ for each edge $e \in E$ such that $f_x(v) = b_v$ for all $v \in V$, and $0 \leq x_e \leq u_e$. This is called a generalized flow. In class, we looked at a special case where all edges are between a vertex with $b_v \leq 0$ and a vertex with $b_v \geq 0$ (the factory-warehouse problem). Now we are considering the problem on a more general digraph.

Solution: Make a new digraph $G' = (V \cup \{r, s\}, E \cup E_{new})$ with two additional vertices, a source $r$ and a sink $s$. Put additional edges from $r$ to $v$ of capacity $-b_v$ if $b_v < 0$, and an edge from $v$ to $s$ of capacity $b_v$ if $b_v > 0$. Call this set of edges $E_{new}$. Then, if the max flow value is equal to $\sum_{v: b_v > 0} b_v$, then we have a feasible solution to the original problem. Otherwise, if the max flow is less than this value, there is no feasible solution (clearly, the max flow cannot be larger than this value). This claim can be proved by first observing that any feasible flow to the original problem gives a max flow of value $\sum_{v: b_v > 0} b_v$, by putting a flow equal to capacity on all edges in $E_{new}$. Secondly, if we have a max flow of value $\sum_{v: b_v > 0} b_v$, then each of the edges in $E_{new}$ have to be at full capacity, thus giving a feasible solution to the original problem, using the flow constraints.

ii. For any subset $S \subseteq V$, let $b(S) = \sum_{v \in S} b_v$. Prove that there is a feasible solution to the generalized flow problem if and only if $b(V) = 0$ and $b(S) \leq u(\delta(V \setminus S))$ for every nonempty $S \subseteq V$.

Solution: The necessity of the condition is easy to verify by simply considering the sum $\sum_{v \in S} f_x(v)$ which should equal $b(S)$.

For the sufficiency, we use the digraph constructed in part i. We show that the min cut value is $\sum_{v: b_v > 0} b_v$, and so there exists a max flow of this value by the max flow-min cut theorem. Then by the argument in part i., we will be done. Consider any $(r, s)$-cut $R$ in $G'$. Let $S = V \setminus R$. Then the total capacity of the cut $R$ in $G'$ is

$$u(\delta(R)) = \sum_{e \in E_{new}} u_e + \sum_{e = vw, v \in V \cap R, w \in V \setminus R} u_e.$$

$$= \sum_{b_v < 0, v \in V} (-b_v) + \sum_{b_v > 0, v \in R} b_v + \sum_{e = vw, v \in V \setminus S, w \in S} u_e.$$

$$= \sum_{v \in S} (-b_v) + \sum_{v \in V, b_v > 0} b(v) + u(\delta(V \setminus S)) - \sum_{b_v < 0, v \in V} b_v = -b(S) + \sum_{v \in V, b_v > 0} b(v) + u(\delta(V \setminus S)).$$

Since $b(S) \leq u(\delta(V \setminus S))$, we have that $u(\delta(R)) \geq \sum_{v \in V, b_v > 0} b(v)$. Of course the cut $R = V \cup \{r\}$ has value $\sum_{v \in V, b_v > 0} b(v)$ in $G'$, so this is the value of the min cut.

2. (Open Pit Mining) In the design of an open pit mine, the region under consideration is divided into 3D blocks. Then decisions are made as to which blocks will be excavated. The only constraint is that if a block needs to be excavated, then any block “above” it also needs to be excavated (with a suitable definition for “above”). This leads to the following abstraction: Consider a digraph $G = (V, E)$ where each node had an associated weight (benefit) $b_v \in \mathbb{R}$ (not necessarily nonnegative). We want to find a subset of nodes to maximize our benefit. However, if choose a node $v$ and there is an edge $vw \in E$, then $w$ must also be chosen. So the goal is to find a subset of vertices $S \subseteq V$ such that $\delta(S) = \emptyset$ and the total benefit $b(S)$ is maximized.
Give an efficient algorithm to solve this problem.

Solution: Make a new digraph $G' = (V \cup \{r, s\}, E \cup E_{new})$ with two additional vertices, a source $r$ and a sink $s$. Put additional edges from $r$ to $v$ of capacity $b_v$ if $b_v > 0$, and an edge from $v$ to $s$ of capacity $-b_v$ if $b_v < 0$. Call this set of edges $E_{new}$. Put a capacity of $\infty$ on all other edges.

Now solve the min cut problem on this graph. Let $R^* = S^* \cup \{r\}$ be a minimum cut in $G'$, where $S^* \subseteq V$. Clearly, no edge from $E$ belongs to $\delta(R^*)$ since they are of infinite capacity, and there is a cut of value $\sum v : b_v > 0 b_v$. We claim that the optimal set of nodes is $S^*$. To show this consider any $(r, s)$-cut $R = S \cup \{r\}$. Then,

$$u(\delta(R)) = \sum_{v \in V \setminus S, b_v > 0}(b_v) + \sum_{v \in S, b_v < 0}(-b_v)$$

$$= \sum_{v \in V \setminus S, b_v > 0}b_v + \sum_{v \in S, b_v > 0}b_v - \sum_{v \in S, b_v < 0}b_v + \sum_{v \in S, b_v < 0}(-b_v)$$

$$= \sum_{v : b_v > 0}(-b_v) + \sum_{v \in S}(-b_v)$$

Since, the first term of the right hand side of the last equation is a constant, minimizing $u(\delta(R))$ is equivalent to maximizing $b(S) = \sum_{v \in S} b_v$.

3. (Problem 3.16 from textbook) Find the maximum matching and minimum cover in the following bipartite graph.

Matching edges are in green, vertex cover nodes are circled in red

4. (Problem 3.17 from textbook) We are given a bipartite graph $G = (V, E)$ with nonnegative weights $b_v$ on vertices $v \in V$. Give an algorithm to find the minimum weight vertex cover for $G$.

Solution: Let $V_1$ and $V_2$ be the partition of the vertices of the bipartite graph $G$. Construct a digraph $G'$ whose vertices are the vertices of $G$ with two additional vertices, a source $r$ and a sink $s$. Every edge in $G$ is directed from $V_1$ to $V_2$ with capacity $\infty$. Additional edges are added from $r$ to every vertex $v$ in $V_1$ with edge capacity $b_v$. Also, new edges are added from all vertices $v$ in $V_2$ to $s$ with edge capacity $b_v$. Let $R$ be any $(r, s)$-cut in $G'$ such that $\delta(R)$ is of finite total capacity. Let $A_1 = V_1 \setminus R$ and $A_2 = V_2 \cap R$. Then,

**Claim 1.** $A = A_1 \cup A_2$ is a vertex cover of weight $\delta(R)$.

**Proof.** We first show that $A$ is a vertex cover. Since every edge from $V_1$ to $V_2$ is of infinite capacity, there are no edges from $R \cap V_1$ to $R \setminus V_2$ since $R$ is of finite capacity. Thus, every edge in $G$ is incident on either a vertex in $R \cap V_2 = A_2$ or $R \setminus V_1 = A_1$ (or both), thus proving $A_1 \cup A_2$ is a vertex cover.

Next, observe that $\delta(R) = D_1 \cup D_2$ where $D_1$ are all edges in $\delta(R)$ from $r$ to $V_1$, and $D_2$ are all edges in $\delta(R)$ from $V_2$ to $s$. Thus, $D_1$ is the set of edges from $r$ to $A_1$ and $D_2$ is the set of all edges from $A_2$ to $s$. Since the total capacity of $D_1$ is the total weight of the vertices in
6. (Problem 3.22 from textbook) We have a family ($A_1$ similarly for $D_2$ and $A_2$), the total weight of $A$ is exactly equal to the total capacity of $\delta(R)$. □

Thus, finding the minimum cut in $G'$ solves the minimum vertex cover problem in $G$.

5. (Problem 3.18 from textbook) Let $G = (V = V_1 \cup V_2, E)$ be a bipartite graph and a nonnegative integer $d_v$ associated with each vertex $v \in V$. Find a polynomial time algorithm to decide if there exists a subset of edges $E' \subseteq E$ such that for every vertex $v$, there are exactly $d_v$ edges incident on it.

Solution: Construct a digraph $G'$ whose vertices are the vertices of $G$ with two additional vertices, a source $r$ and a sink $s$. Every edge in $G$ is directed from $V_1$ to $V_2$ with capacity 1. Additional edges are added from $r$ to every vertex $v$ in $V_1$ with edge capacity $d_v$. Also, new edges are added from all vertices $v$ in $V_2$ to $s$ with edge capacity $d_v$.

We claim the problem has a feasible solution if and only if the maximum $r,s$ flow in $G'$ is $\sum_{v \in V_1} d_v = \sum_{v \in V_2} d_v$. Indeed, $\sum_{v \in V_1} d_v$ is the maximum possible value of a flow because the cut consisting only of the vertex $s$ has this capacity. If the problem has a feasible solution, then simple put flow values of $d_v$ on all the edges incident on $r$ and $s$, and put flow value 1 on all the edges used in the problem. Not the other hand, if the maximum flow value is $\sum_{v \in V_1} d_v$, consider a max flow solution which takes only integral flow values, which we know exists by results from class. Then the set of edges in the original graph $G$ which have positive flow values in $G'$ form a set of edges satisfying the property. This is because by the integrality property, these flow values are 1 and thus we have exactly $d_v$ edges at every vertex because of the flow constraints.

6. (Problem 3.22 from textbook) We have a family $(S_1, \ldots, S_k)$ of subsets of a set $Q$. A system of distinct representatives (SDR) is a set of distinct elements $\{q_1, \ldots, q_k\} \subseteq Q$ such that $q_i \in S_i$ for all $i = 1, \ldots, k$. Prove that a family has an SDR if and only if for every subset $I$ of $\{1, \ldots, k\}$ we have $|\bigcup \{S_i : i \in I\}| \geq |I|$.

Solution: Construct a bipartite graph $G$ where $V_1$ is a set of $k$ nodes corresponding to $S_1, \ldots, S_k$, and $V_2$ is the set of all elements in $Q$. We put an edge $ij$, $i \in V_1, j \in V_2$ if $j \in S_i$. Now a matching $M$ of size $k$ gives an SDR by letting $q_1, \ldots, q_k$ be the $M$-covered vertices in $V_2$. Thus, we have an SDR if and only if there is a matching of size $k$, i.e., one that saturates $V_1$. Also, notice that if there is a matching of size $k$, then the condition “for every subset $I$ of $\{1, \ldots, k\}$ we have $|\bigcup \{S_i : i \in I\}| \geq |I|$” must be satisfied. So need to prove the converse: we assume that the maximum matching in $G$ is strictly less than $k$ and show there exists an $I^*$ that violates this condition.

Construct a directed graph $\tilde{G}$ with capacities from $G$ by doing the following. The vertices of $\tilde{G} = V_1 \cup V_2 \cup \{r, s\}$. The edges of $G$ are included in $\tilde{G}$ with a direction from $V_1$ to $V_2$ and are given a capacity of $\infty$. Add a directed edge from $r$ to each vertex in $V_1$ with capacity 1. Add a directed edge from each vertex in $V_2$ to $s$ with capacity 1. The maximum matching size in $G$ is equal to the maximum flow in $\tilde{G}$. If the maximum matching in $G$ is strictly less than $k$, then by the max-flow min-cut theorem, there exists a $(r, s)$-cut $R$ of value strictly less than $k$. Now since the original edges have capacity $\infty$, all edges incident on $R \cap V_2$ must be incident on $R \cap V_1$, so if let $I^* = R \cap V_1$, then $\bigcup \{S_i : i \in I^*\} \subseteq R \cap V_2$. Also, notice that the $\delta(R) = (k - |R \cap V_1|) + |R \cap V_2|$ which we know is strictly less than $k$. So, $(k - |R \cap V_1|) + |R \cap V_2| < k$ which implies $|R \cap V_2| < |R \cap V_1|$. But then $|\bigcup \{S_i : i \in I^*\} \leq |R \cap V_2| < |R \cap V_1| = |I^*|$ contrary to the hypothesis.

7. (Problem 3.23 from textbook) Suppose every node of a bipartite graph has degree $p \geq 1$. Show that there exists a matching that covers every vertex. Such a matching is called a
8. (Problem 3.26 from textbook) For families \( (S_1, \ldots, S_k) \) and \( (T_1, \ldots, T_k) \) of subsets of \( Q \), a common SDR is a set \( \{q_1, \ldots, q_k\} \subseteq Q \) that is an SDR for both families. Prove that there exists a common SDR if and only if for every pair \( I, J \) of subsets of \( \{1, \ldots, k\} \) we have

\[
|\bigcup_i (S_i : i \in I) \cap \bigcup_j (T_j : j \in J)| \geq |I| + |J| - k.
\]

[Hint: Set up a max-flow model to solve the common SDR problem, and use the max flow-min cut theorem]

Solution: Consider a digraph \( G \) with \( V = V_1 \cup V_2 \cup V_3 \cup V_4 \cup \{r, s\} \), where \( V_1 = V_4 = \{1, \ldots, k\} \), and \( V_2 = V_3 = Q \). We first put edges from \( r \) to all vertices in \( V_1 \) of capacity 1. For every \( i \in V_1 \), \( j \in V_2 \) such that \( j \in S_i \), we put the directed edge \( ij \) of capacity \( \infty \). For every \( j \in Q \), we put an edge from \( i \in V_2 \) to \( i \in V_3 \) of capacity 1. For every \( j \in V_3 \) and \( i \in V_4 \) such that \( j \in S_i \), we put a directed edge \( ji \) of capacity \( \infty \). Finally, there is a directed edge from every vertex in \( V_4 \) to \( s \) of capacity 1.

It can be shown that there is a flow of size \( k \) in this graph if and only if there exists a common SDR (Amitabh: I am leaving out the details here, but one should prove this claim).

We now prove that there exists a common SDR if and only if for every pair \( I, J \) of subsets of \( \{1, \ldots, k\} \) we have

\[
|\bigcup_i (S_i : i \in I) \cap \bigcup_j (T_j : j \in J)| \geq |I| + |J| - k.
\]

\((\Rightarrow)\) Let \( C \) be the common SDR. For any subset \( I, J \subseteq \{1, \ldots, k\} \), let \( C_I \subseteq C \) denote the representative of \( S_i, i \in I \), and \( C_J \subseteq C \) denote the representative of \( T_j, j \in J \). Certainly, \( |C_I| = |I| \) and \( |C_J| = |J| \) since \( C \) is a common SDR. Moreover, \( (C_I \cap C_J) \subseteq (\bigcup_i (S_i : i \in I)) \cap (\bigcup_j (T_j : j \in J)) \) and \( C_I \cup C_J \subseteq C \). Thus, \( |(C_I \cap C_J)| \leq |(\bigcup_i (S_i : i \in I)) \cap (\bigcup_j (T_j : j \in J))| \) and \( |C_I \cup C_J| \leq |C| \). Thus,

\[
|(\bigcup_i (S_i : i \in I)) \cap (\bigcup_j (T_j : j \in J))| + k = |(\bigcup_i (S_i : i \in I)) \cap (\bigcup_j (T_j : j \in J))| + |C|
\geq |C_I \cap C_J| + |C_I \cup C_J|
= |C_I| + |C_J|
= |I| + |J|.
\]

\((\Leftarrow)\) Let \( (S_1', \ldots, S_k') \) be a common SDR. Consider any subset \( I, J \subseteq \{1, \ldots, k\} \). We define \( r' = |S_1'| \). We now prove that there exists a common SDR if and only if for every pair \( I, J \) of subsets of \( \{1, \ldots, k\} \) we have
(⇐) We prove the contrapositive. Suppose we do not have a common SDR. Then the max flow value in G is strictly less than k and so is the minimum cut value. Let \( R^* \) be the minimum cut. Let \( I^* = R^* \cap V_1 \) and \( J^* = V_4 \setminus R^* \). Let \( V^+ = R^* \cap V_2 \) and \( V^- = V_3 \setminus R^* \). Since the min cut value is strictly less than k, \( \delta(R^*) < k \). Since \( \delta(R^*) \) is of finite total capacity, the only edges in \( \delta(R^*) \) are from \( V_1 \setminus I^* \), from \( V^+ \) to \( V^- \), and from \( V_4 \cap R^* \) to \( s \). Thus,

\[
\delta(R^*) = (k - |I^*|) + |V^+ \cap V^-| + (k - |J^*|) < k
\]

\[
\Rightarrow |V^+ \cap V^-| < |I^*| + |J^*| - k.
\] (1)

Furthermore, there are no edges from \( I^* \) to \( V_2 \setminus V^+ \), and no edges from \( J^* \) to \( V_3 \setminus V^- \). Thus, \( \bigcup(S_i : i \in I^*) \subseteq V^+ \) and \( \bigcup(T_j : j \in J^*) \subseteq V^- \). Hence \(|\bigcup(S_i : i \in I) \cap \bigcup(T_j : j \in J)| \leq |V^+ \cap V^-|\). Combining with (1), we get

\[
|\bigcup(S_i : i \in I) \cap \bigcup(T_j : j \in J)| \leq |V^+ \cap V^-| < |I^*| + |J^*| - k
\]

which is a contradiction to the assumption.

9. (Problem 3.29 from textbook) Projects \( P_1, P_2, \ldots, P_k \) are available to be undertaken, and there is a set of resources \( R_1, \ldots, R_\ell \) that can be used to complete these projects. With each project \( i \) that is undertaken, we get a nonnegative revenue \( r_i \). Each project \( i \) requires a set \( S_i \subseteq \{ R_1, \ldots, R_\ell \} \) of resources to be available, and each resource \( R_j \) has an associated cost \( c_j \).

However, if resource \( R_j \) is purchased, it is available for any project for which it is required. Give an algorithm to choose a set of projects (and the required set of resources to buy) to maximize the revenue minus the cost.

Solution: We set up a directed graph \( G = (V, E) \) where \( V \) has a vertex for each project, which we denote by \( P_i, i = 1, \ldots, k \); a vertex for each resource, which we denote by \( R_j, j = 1, \ldots, \ell \); and two additional vertices \( r \) and \( s \). We add edges from \( r \) to each vertex \( P_i \) with capacity \( r_i \) for \( i = 1, \ldots, k \). We next add edges from each resource \( R_j \) to \( s \) of capacity \( c_j \) for \( j = 1, \ldots, \ell \). Finally we have edges from \( P_i \) to each resource vertex in the set \( S_i \) from the problem statement, with capacity \( \infty \).

**Claim 2.** If \( R^* \) is minimum \((r, s)\)-cut in this graph, then \( R^* \cap \{P_1, \ldots, P_k\} \) is the optimal set of jobs to choose.

**Proof.** Let \( R \) be any \((r, s)\)-cut. Let \( I \subseteq \{1, \ldots, k\} \) such that \( R \cap \{P_1, \ldots, P_k\} = \{J_i : i \in I\} \). Since the min cut is finite (we can choose \( R = \{r\} \)), there can be no edge of \( \infty \) capacity in \( \delta(R) \) and thus \( J := \bigcup_{i \in I}(S_i) \subseteq R \cap \{R_1, \ldots, R_\ell\} \). Thus,

\[
\delta(R) \geq \sum_{i \in I} r_i + \sum_{j \in J} c_j = \sum_{i=1}^{k} r_i + \left( \sum_{j \in J} c_j - \sum_{i \in I} r_i \right)
\]

This relation shows that \( \delta(R) \geq C + \left( \sum_{j \in J} c_j - \sum_{i \in I} r_i \right) \) where \( C \) is the constant \( \sum_{i=1}^{k} r_i \). Thus, for the minimum cut \( R^* \), if we define \( I^* \) such that \( R^* \cap \{P_1, \ldots, P_k\} = \{P_i : i \in I^*\} \), \( \delta(R^*) \geq C + \left( \sum_{j \in J^*} c_j - \sum_{i \in I^*} r_i \right) \) where \( J^* := \bigcup_{i \in I^*}(S_i) \). Note that if we choose the set of jobs indexed by \( I^* \), then it suffices to get the resources indexed by \( J^* \). And thus our total cost equals \( \sum_{j \in J^*} c_j - \sum_{i \in I^*} r_i \). This shows that

\[
\delta(R^*) - C \geq \sum_{j \in J^*} c_j - \sum_{i \in I^*} r_i \geq \min_{J \subseteq \{1, \ldots, k\}} \left( \sum_{j \in J \setminus \bigcup_{i \in I}(S_i)} c_j - \sum_{i \in I} r_i \right). \] (2)
On the other hand, for any set of projects indexed by $I \subseteq \{1, \ldots, m\}$, define the $(r,s)$-cut $R(I) = \{r\} \cup \bigcup_{i \in I} \{P_i\} \cup \bigcup_{j \in \cup_{i \in I}(S_i)} \{R_j\}$. Observe that $\delta(R(I)) - C = \sum_{j \in \cup_{i \in I}(S_i)} c_j - \sum_{i \in I} r_i$. Thus

$$\min_{I \subseteq \{1, \ldots, m\}} \left( \sum_{j \in \cup_{i \in I}(S_i)} c_j - \sum_{i \in I} r_i \right) = \min_{I \subseteq \{1, \ldots, k\}} (\delta(R(I)) - C) \geq \delta(R^*) - C. \quad (3)$$

Combining (2) and (3), we get the result. \qed