Assignment 1

ISYE 7686 - (Discussed with Qiaomei Li and Xiaofan Yuan)

January 25, 2020

Due Date: March 1, 2019.

In preparing solutions for this assignment, please abide by the Georgia Tech academic honor code. We will not tolerate copying of answers. Discussions are encouraged but the final answer must be based upon your understanding of the question. Also write the names of the people you discussed the questions with.

Q1. Prove the following:

(i) If a matrix A satisfies the Ghouila-Houri condition (i.e. every collection of columns of A can be split in to two parts such that sum of columns in one part minus the sum of columns in other part is a vector with entries in 0, ± 1), then it is totally unimodular.

First, taking a collection of columns to be a single column, gives that all columns are vectors with every entry 0, 1, or -1. Now, we wish to show that every square subdeterminant of A is 0, 1, or -1.

I ask: if A (an m by n matrix) satisfies the Ghouila-Houri condition, is it true that any square submatrix of A satisfies the Ghouila-Houri condition? I claim yes. Take a square submatrix A' of A given by $A' := A^{\{i_1,i_2,\ldots,i_k\}}_{\{j_1,j_2,\ldots,j_k\}}$. Now, we wish to show that for any subset of the columns of A', there exists a bipartition of these columns such that the sum of the columns in one part minus the sum of the columns in the other part is a $\{0,1,-1\}$ vector. So, take some subset $J \subseteq \{j_1, j_2, \ldots, j_k\}$ of the columns of A'. The set $J \subseteq [n]$ is also a subset of the columns of A, which means that the set of columns $\{A_j | j \in J\}$ can be paritioned into two sets B_1, B_2 of column indices such that $\sum_{j\in B_1} A_j - \sum_{j\in B_2} A_j \in \{0,1,-1\}^m$. Now, $\sum_{j\in B_1} A_j - \sum_{j\in B_2} A_j \in \{0,1,-1\}^m$ means that $\sum_{j\in B_1} A_j^i - \sum_{j\in B_2} A_j^i \in \{0,1,-1\}$ for all row indices $i \in [m]$. IN particular, this holds for all row indices $i \in \{i_1, i_2, \ldots, i_k\}$ (the rows of A') and we are done.

Thus, it suffices to prove that any square matrix A satisfying the Ghouila-Houri condition is unimodular, meaning A itself has determinant 0,1, or -1 because any square submatrix also satisfies the Ghouila-Houri condition.

So, say that A is a square matrix satisfying this condition. We wish to show $\det(A) \in \{0,1,-1\}$. Clearly, $\det(A^T) = \det(A)$. Thus, it suffices to prove that $\det(A^T) \in$

$\{0, 1, -1\}$. We know that row reducing a matrix changes the determinant as follows. Multiplying a row by c multiplies the determinant by c. Swapping rows multiplies the determinant by -1. Adding a scalar multiple of one row to another row does not change the determinant. So, one row reduces as follows:

- Step 1: let j_0 be the first column for which there is some entry $(A^T)^{i_0}_{j_0} =\in \{1, -1\}$. If no, such j_0 exists, then A^T is the zero matrix and therefore has determinant 0 and we are done. Now, if $(A^T)^{i_0}_{j_0} = -1$, then multiply row i_0 by -1 to get the matrix we'll call B that we update through row operations. If $(A^T)^{i_0}_{j_0} = 1$, do nothing. Now (in either case), swap rows 1 and i_0.
- Step 2: Now, for all rows $i \neq i_0$, we know that the set of rows $\{i, 1\}$ of B (as modified so far) (which were originally columns of A) can be partitioned such that the difference of the sum of one part and the sum of the other part is a $\{0, 1, -1\}$ row vector. Now, if $B^i_{j_0} = 0$ we are happy, because this means that the ith row already has a zero below the pivot at $B^1_{j_0}$ (and also by choice of j_0 all entries to the left of $(A^T)^i_{j_0}$ are zero). Otherwise, if $B^i_{j_0} = -1$, multiply B^i by -1 to get the new B. Otherwise, if $B^i_{j_0} = 1$, do nothing. Now, we know that either ($B^i_{j_0} - B^1_{j_0} = 0$ and $B^i - B^1$ is $\{0, 1, -1\}$ row vector) or ($B^1_{j_0} - B^i_{j_0} = 0$ and $B^1 - B^i$ is $\{0, 1, -1\}$ row vector). Why? Well, the fact that $B^1_{j_0} = B^i_{j_0} = 1$ means that neither part of the partition of $\{i, 1\}$ can be empty since otherwise the sum of the other part (having both row vectors) would have 2 in the first coordinate, contradicting the Ghouila-Houri condition. So, each part has exactly one of $\{1, i\}$ giving what I just wrote above. Now, if ($B^i_{j_0} - B^1_{j_0} = 0$ and $B^i - B^1$ is $\{0, 1, -1\}$ row vector) (the first case), then we update B by replacing $B^i := B^i - B^1$ (which does not change $\det(B)$). If ($B^1_{j_0} - B^i_{j_0} = 0$ and $B^1 - B^i$ is $\{0, 1, -1\}$ row vector) (the second case), then one replaces $B^i := B^1_{j_0} - B^i_{j_0} = 0$ which multiplies $\det(B)$ by -1. One repeats this process until B has zeros below the pivot $B^1_{j_0}$.
- Step 3: Then, consider the matrix $B := B^{[2:n]}_{[j_0+1:m]}$. Repeat steps 1-3 until B is empty (meaning one has run out of columns or rows) (of course while row reducing some smaller B that we have recursed on, we think of this as part of our original n by m B and since everything to the left of this smaller B we are recursing on is zero, this part as well as the part above our smaller B is unaffected).

So, such a procedure produces a matrix B row equivalent to A^T in row echelon form which has determinant $\det(B) = \pm\det(A^T) = \pm\det(A)$. Why? Because of the Ghouila-Houri condition, we were able to reduce to row echelon form without ever multiplying a row by any c other than 1 or -1. Now, the matrix B is upper triangular since any square matrix in row echelon form is upper triangular. Thus, the determinant is the product of the elements on the diagonal which are all 0 or 1. Thus, $\det(B) \in \{0, 1\}$, which means that $\det(A^T) = \det(A) \in \{0, 1, -1\}$ and we are done. (Remark: we didn't actually need to multiply any rows by -1 if we didn't want to, we still know that as before for the set of row indices $1, i$ there exists a partition such that the sum of the rows in one part minus the sum of the rows in the other part is a $\{0, 1, -1\}$ vector, which still gives us the desired zero-ing out under pivots to get an upper triangular matrix without ever explicitly multiplying a row by -1).

(ii) Let A be a matrix with entries in 0, 1 such that in every row the ones occur in consecutive columns, then A is totally unimodular. We first note that any square submatrix of a matrix A satisfying the conditions of this problem also satisfies the conditions of this problem, so it suffices to prove that A is unimodular, which means that $\det(A) = \det(A^T) = \pm 1$. So, it suffices to show that $\det(A^T)$ is unimodular (or even better totally unimodular). According to the previous part, it suffices to show that for any subset S of the column indices of $B := A^T$, there is some partition of S such that the sum of the columns whose indices lie in one part minus the sum of the columns whose indices lie in another part is a $\{0, 1, -1\}$ vector. It actually suffices to prove the statement for the case where $S = [n]$ so that our subset of column indices is all the column indices. Otherwise, we know that the proper subset of columns defines a submatrix B' of B where B' also satisfies the conditions of the problem (well it's transpose does) and the set S' corresponds to all columns of B' so if we proved the statement for the case where the subset is all of the columns we are done in this case as well. We will prove by induction on the number of columns in such a subset S that such a partition exists. Base case: $|S| = 1$, then $S = \{j\}$. The partition $\{\{j\}, \emptyset\}$ is the desired partition. Now for the inductive step assume that $|S| \geq 2$. If there exist $j, j' \in S$ such that the row indices in which 1's appear in column j are a subset of the row indices in which 1's appear in column j' and either the first row in which a 1 appears in columns j and j' are the same or the last row in which a 1 appears in columns j and j' are the same, then one can delete j, j' from S instead consider the set S' of columns whose indices are in S as well as the new column vector $B_j - B_{j'}$ which is a $\{0, 1\}$ vector in which the 1's appear in consecutive rows and by induction there is a partition S'_1, S'_2 of this set of columns into two parts such that the sum of columns in the first part minus the sum of the columns in the other part is a $\{0, 1, -1\}$ vector. Now, WLOG the column $B_j - B_{j'}$ (call it's index j^*) appears in the first part which means that one can then create the desired partition of S by setting $S = (S'_1 \setminus \{j^*\} \cup \{j\}) \sqcup (S'_2 \cup \{j'\})$ and we are done. So, now assume that for all $j, j' \in S$, one does not have that the row indices at which 1's appear in column j is contained in the row indices at which 1's appear in column j' and that the 1's start in the same row for both columns or end in the same row for both columns. So, if there are never two columns such that their 1's start at the same row because then their row indices with 1's would satisfy that one set is contained in the other. So, for all pairs of column indices j, j' their 1's start at different rows and end at different rows. Sort the columns by the row in which their 1's start so that the leftmost column has it's 1's starting at the earliest row index. Now, this matrix is lower triangular, meaning that it's determinant is the product of the diagonal entries so the determinant of this matrix is either 1 or 0. Since the original matrix was some permutation (of these columns) the determinant of the original matrix was $\{0, 1, -1\}$ and we are done.

Q2. Two persons are playing the following game on a graph. One after another the players choose vertices (one per turn) $v_1, v_2, v_3, \dots$ so that v_i is adjacent to v_{i+1} for all $i \geq 0$. The last player which is able to choose a vertex wins. Prove that the first player has a winning strategy if and only if the graph has no perfect matching.

We first show that if the graph has a perfect matching then player one has no winning strategy. For any vertex player 1 picks, player 2 has a strategy such that player 2 has a strategy such that player 2 always wins no matter what player 1 does. Namely, player 1 picks a vertex, v_1, at move 1. Then, player 2 picks the vertex, v_2, adjacent to it in the chosen fixed perfect matching M (which exists by assumption). Now, either there are no vertices yet to be chosen adjacent to v_2 in which case player 2 wins, or player 1 chooses some vertex, v_3, adjacent to v_2 in G. Then player 2 continues this strategy. Namely, say that player 1 has just chosen vertex $v_i =: u$ (with i odd) (we give it the additional name u for reasons that will become clear soon). Then, player 2 chooses the vertex $v_{i+1} =: w$ which is adjacent to v_i in M.

We show that this strategy is valid (namely possible to execute for player 2) by induction. Say that player 1 has just chosen up to v_i and now player 2 has to choose v_{i+1}, and that up until and including move $i-1$ player 2 has always been able to chose the vertex matched to player 1's previous pick (really the only thing that could go wrong is that this vertex has already been chosen so we are showing that that can't happen by induction). We now wish to show that player 2's current pick $v_{i+1} = w$ has not yet been chosen. Well if it has, it has either been chosen by player 1 or player 2.

If w has already been chosen by player 1 at some earlier move (move k) as v_k for $k < i$ odd (by the fact that player 1 moves on odd turns), then at the next move player 2 would have chosen u (actually named as v_{k+1} since it would have been chosen at move $k+1$). Also, $k < i$ with k and i both odd implies that actually, $k \leq i-2$, which then implies that $k+1 \leq i-1$ so that player 2 would have chosen u at step $k+1 \leq i-1$, a contradiction as that means when player 1 chose u, player 2 already chose it earlier.

Now, what if w was already chosen by player 2 at some earlier move? If so, then there was some even $k \leq i$ (however k even and i odd implies that actually $k \leq i-1$) for which $v_k = w$. Why did player two pick $v_k = w$? According to player 2's strategy, it chose $v_k = w$ because it was adjacent to v_{k-1} in M, which in particular means that $v_{k-1} = u$, a contradiction because then player 1 chose the vertex u twice.

So, player 1 loses in this case.

Next, we wish to show that if G has no perfect matching, then player 1 has a winning strategy. The fact that G has no perfect matching means that the maximum matching in G has size $|M| < \frac{|V(G)|}{2}$. Consider such a maximum matching.

Lemma 1: If $M \subseteq E(G)$ is a maximum matching in G then $W := \{v \in V(G) : v \text{ not covered by } M\}$, then W is an independent set in G.

Proof: Otherwise there exists some edge $uw \in E(G)$ such that $u, w \in W$, but then adding the edge uw to M results in a larger matching, a contradiction.

Lemma 2: If $M \subseteq E(G)$ is a maximum matching in G and $W := \{v \in V(G) : v \text{ not covered by } M\}$, then there does not exist a sequence of vertices $w_1, w_n \in W$, $u_2, u_3, \ldots, u_{n-1} \in \bar{W}$ such that $w_1u_2, u_{n-1}w_n \in E(G)$ and $u_iu_{i+1} \in M$ for all i even $\in [n-1]$.

Proof: Otherwise, $w_1, u_2, u_3, \dots, u_{n-1}, w_n$ is an M augmenting path.

Now, we show that if G has no perfect matching then player 1 has the following winning strategy. Let M be a maximum matching in G and $W := \{v \in V(G) : v \text{ not covered by } M\}$. Then, player 1 picks an arbitrary $v_1 = w \in W$ which exists because by assumption $W \neq \emptyset$. Then, either w is isolated, in which case player 1 wins, or it is not, in which case player 2 has a valid move as there is some edge $w'w$ incident with w. Now, because W is an independent set, necessarily, one has that $w' \in W$ so that namely, player 2's next pick (no matter what it is) $v_2 = w'$ is covered by M. Thus, player 1 then picks the vertex v_3 adjacent to v_2 in M. Then, either there are no unpicked vertices w'' adjacent to v_3 in G, in which case player 1 wins, or there is some vertex w'' adjacent to v_3 in G which player 2 picks. By lemma 2, $w'' \in \bar{W}$. So, $v_4 = w'' \in \bar{W}$, and then once again player 1 picks the vertex matched with it in M. Continuing in this fashion, player 1 wins.

Q3. Let $D = (V, A)$ be a directed graph and $\mathcal{C}$ denote the set of cycles in D. Let $\chi(C) \in \{0,1\}^A$ denote the indicator vector of cycle C. Show that non-negative circulations form the cone generated by $\{\chi(C) : C \in \mathcal{C}\}$.

Well, clearly $\text{cone}(\{\chi(C) : C \in \mathcal{C}\}) \subseteq [$ set of nonneg circulations $]$ since $\{\chi(C) : C \in \mathcal{C}\} \subseteq [$ set of nonneg circulations $]$ and for any set of non-negative circulations $\{f_i\}$ one has that the the function $g = \sum_{i \in I} a_i f_i : E \to \mathbb{R}$ (with $a_i \in \mathbb{R}^{\geq 0}$) is first of all non-negative and is also a circulation since the sum of circulations is a circulation.

Proof: $f_i : E \to \mathbb{R}$ a circulation for all $i \in I$ means that flow conservation is satisfied for all $i \in I$, namely $\sum_{e \in \delta^+(v)} f_i(e) - \sum_{e \in \delta^-(v)} f_i(e) = 0$. Then, by definition of $g(e) = \sum_{i \in I} a_i f_i(e)$, (with $a_i \in \mathbb{R}^{\geq 0}$) one has that

$$\begin{aligned}
\sum_{e \in \delta^+(v)} g(e) - \sum_{e \in \delta^-(v)} g(e) &= \sum_{e \in \delta^+(v)} \sum_{i \in I} a_i f_i(e) - \sum_{e \in \delta^-(v)} \sum_{i \in I} a_i f_i(e) \\
&= \sum_{i \in I} a_i(\sum_{e \in \delta^+(v)} f_i(e) - \sum_{e \in \delta^-(v)} f_i(e)) \\
&= \sum_{i \in I} 0 \qquad = 0.
\end{aligned}$$

So, It remains to show that for all $f : A \to \mathbb{R}_{\geq 0}$ there exist $\{a_c \in \mathbb{R}_{\geq 0} | c \in \mathcal{C}\}$ such that $f = \sum_{c \in \mathcal{C}} a_c \chi(c)$. Ok, now, the fact that f_i is a circulation means that flow conservation is satisfied. We construct an algorithm to decompose f into the non-negative sum of indicator vectors of (directed) cycles.

1. Order the elements of $\mathcal{C}$ so that $\mathcal{C} = \{c_i : i \in [|\mathcal{C}|]\}$.

2. $f^1 := f$

3. For $i \in |\mathcal{C}|$:

(a) let $\mathcal{C}' := \{c_i \in \mathcal{C}' : f^i(a) > 0 \text{ for all } a \in E(c)\}$ (we are removing the cycles that now have zero flow on some edge).

(b) So, take $c \in \mathcal{C}'$. Name it $d_i := c$. Now, find

$$k_i := \min_{a \in A \cap c} f^i(a) > 0.$$

(c) Let $f^{i+1}(e) = f^i(e) - k_i$ for all $e \in c$ and $f^{i+1}(e) = f^i(e)$ for all $e \in A \setminus c$.

Then, I claim that this algorithm correctly computes $f = \sum_{i \in |\mathcal{C}|} k_i \chi(c_i)$.

Proof: First, we show that at termination, the leftover flow is zero on all edges, namely we show that $f^{|\mathcal{C}|+1} \equiv 0$ or $f^{|\mathcal{C}|+1}(a) = 0$ for all $a \in A$. We prove so by induction. Namely, we show that at each iteration, there is one less cycle c such that $f^i(a) > 0$ for all $a \in c$, or said in words, that there is at least one less cycle of strictly positive flow at each iteration. Say we are given f^i such that $y = |\{c_i \in \mathcal{C}' : f^i(a) > 0 \text{ for all } a \in E(c)\}|$. Then, at iteration i we find a cycle $c \in C'$ and deaugment its flow by $k_i = \min_{a \in A \cap c} f^i(a) > 0$ (with $a \in \operatorname{argmin}_{a \in A \cap c} f^i(a) > 0$) which then gives that $f^{i+1}(a) = 0$ which in particular means that we then remove c from $\mathcal{C}'$ at the next step leaving at least one less cycle (perhaps there were other cycles in $\mathcal{C}'$ which also contained a, who now also will be removed from $\mathcal{C}'$) of strictly positive flow.

Lemma: The function f^i is a circulation for all i.
Proof: Base case: $f^1 = f$ is a circulation by assumption. Then, inductive step: one notes that $f^i = f^{i-1} + (-k_i)g_i$ where g_i is the circulation defined by $g_i(a) = 1$ for all $a \in d_i$ and $g_i(a) = 0$ for all $a \in A \setminus d_i$. Clearly, since d_i is a cycle, g_i is a valid circulation. Finally, since I showed earlier in this question that the sum of circulations is a circulation, that gives that f^{i+1} is a circulation, concluding the proof.

So, at the termination of this algorithm, one is left with $f^{|\mathcal{C}|+1}$ a circulation such that for all cycles c in G, there is some $a \in A(c)$ such that $f^{|\mathcal{C}|+1}(a) = 0$. I claim that this implies that $f^{|\mathcal{C}|+1} \equiv 0$ is the zero function. Why? Assume not. Assume for contradiction $f^{|\mathcal{C}|+1}(a) > 0$ for some $a \in A$. Now, say $a = a_1 a_2$. Then, flow into $a_2 > 0$ implies that flow out of $a_2 > 0$. So, there exists $a_3 \in V(G)$ such that $a_2 a_3 \in A$ and $f^{|\mathcal{C}|+1}(a_2 a_3) > 0$. Continuing in this fashion, one gets a directed path $a_1, a_2, \dots a_k$ on edges with strictly positive flows. Assuming for contradiction that $a_k \notin \{a_1, a_2, \dots, a_{k-1}\}$, then necessarily there exists $a_{k+1} \in V$ such that $a_k a_{k+1} \in A$ and $f^{|\mathcal{C}|+1}(a_k a_{k+1}) > 0$. So, either $a_k \in \{a_1, a_2, \dots, a_{k-1}\}$ or the path can be extended. Thus, extending the path until one cannot extend anymore (which is possible since G is finite) results in a cycle of strictly positive flow on all edges, a contradiction to the fact that for all cycles c in G, there is some $a \in A(c)$ such that $f^{|\mathcal{C}|+1}(a) = 0$ (this was not assumed for contradiction, this is simply true by construction of our algorithm and the fact that the number of strictly positive flows must decrease at each step). Thus, we see that assuming that some edge a had non zero flow $f^{|\mathcal{C}|+1}(a)$ gave us a contradiction, which means that $f^{|\mathcal{C}|+1}(a) \equiv 0$ is the zero flow.

So,

$$\begin{aligned} f^{|\mathcal{C}|+1} &= f^{|\mathcal{C}|} - k_{|\mathcal{C}|}g_{|\mathcal{C}|} \\ &= (f^{|\mathcal{C}|-1} - k_{|\mathcal{C}|-1}g_{|\mathcal{C}|-1}) - k_{|\mathcal{C}|}g_{|\mathcal{C}|} \\ &= f^1 - \sum_{i=1}^{|\mathcal{C}|} k_i g_i \\ &\equiv 0 \end{aligned}$$

which means that

$$f =: f^1 = \sum_{i=1}^{|\mathcal{C}|} k_i g_i.$$

So, namely f is a conic combination of non-negative circulations g_i (which are the indicator vectors of our cycles), which means that f is in the cone of $\{\chi(c) : c \in \mathcal{C}\}$. So, $\{\text{non-negative circulations}\} \subseteq \text{Cone}(\{\chi(c) : c \in \mathcal{C}\})$. Now, we have both $\{\text{non-negative circulations}\} \subseteq \text{Cone}(\{\chi(c) : c \in \mathcal{C}\})$ and $\text{Cone}(\{\chi(c) : c \in \mathcal{C}\}) \subseteq \{\text{non-negative circulations}\}$, which means that $\{\text{non-negative circulations}\} = \text{Cone}(\{\chi(c) : c \in \mathcal{C}\})$.

Q4. Let G be a graph with vertices $v_1, \ldots, v_n$. Give an algorithm that, given a sequence $d_1, ..., d_n$, decides in polynomial time if G admits an orientation such that $\delta^{out}(v_i) = d_i$ for all $1 \leq i \leq n$.

We construct an auxilliary graph which has a perfect matching exactly when G admits such an orientation. Namely, take $v_i \in G$ with $\deg(v_i) = r_i$, then make r_i copies of v_i each corresponding to an edge incident to v_i. Say the set of edges incident to v_i are $\{v_iv_{i_1}, v_iv_{i_2}, \ldots, v_iv_{i_{r_i}}\}$. Call the r_i copies of v_i $\{v^i_{i_1}, v^i_{i_2}, \ldots, v^i_{r_i}\}$. Do this for all $v_i \in V(G)$ to get the vertex set of the graph $V(G')$ Then, for every edge $v_iv_j \in E(G)$, add the edge $v^i_jv^j_i$ to $E(G')$. Now, for each edge $v^i_jv^j_i \in E(G')$, subdivide (in the graph theoretic definition) it by adding vertex $X_{(i,j)}$ in the middle. Finally, for each vertex v_i in G which is to have out degree d_i in some orientation of G, modify G' as follows, for each v_i create a set S_i of d_i many vertices then add edges to create a complete bipartite graph between S_i and all copies $\{v^iv_{i_1}, v^iv_{i_2}, \ldots, v^iv_{i_{r_i}}\}$ of v_i. Now, I claim that G' has a perfect matching exactly when G admits the desired orientation. Namely, if G' has a perfect matching, then one orients the edges of G as follows. The fact that G' has a perfect matching means that $X_{(i,j)}$ is matched with exactly one of v^i_j or v^j_i. If $X_{(i,j)}$ is matched with v^i_j, we orient the edge $v_iv_j \in E(G)$ from v_j to v_i. Otherwise, if $X_{(i,j)}$ is matched with v^j_i we orient the edge $v_iv_j \in E(G)$ from v_i to v_j. Such an orientation is logically consistent since exactly one of these things happens. Then, one notes that if G' has such a perfect matching, then one must also have that for all $i \in [|V(G)|]$ and all $s \in S_i$, s is matched, so that namely s is matched to some copy of $v^i_{i_k}$ of v_i. Then, s matched to $v^i_{i_k}$ implies that $v^i_{i_k}$ is NOT matched with $X_{(i,i_k)}$, which means that $X_{(i,i_k)}$ is matched with $v^{i_k}_i$, and according to our rule that means we orient the edge v_iv_j from v_i to v_j, giving d_i outgoing edges in total from v_i (one for each $s \in S_i$). We then see that the constructed orientation on G satisfies our condition on the desired out degrees.

So, in order to decide whether such an orientation exists it suffices to determine whether G' has a perfect matching. One can do so by running Edmond's Blossom Algorithm which has runtime $O(n^2m)$. `https://stanford.edu/~rezab/classes/cme323/S16/projects_reports/shoemaker_vare.pdf`

Q5. Prove that a cubic graph has a nowhere zero 3-flow if and only if the graph is bipartite.

We recall that a graph has a nowhere zero 3-flow if and only if it has a nowhere zero $\mathbb{Z}_3$ flow. Thus, we show that a graph cubic graph is bipartite if and only if it has a nowhere zero $\mathbb{Z}_3$ flow.

Lemma:
Say a graph G is r-regular and bipartite. Does G have a perfect matching? Is Hall's condition satisfied? To start one notes that if G has partite sets $X, Y \subseteq V(G)$ then $|E(G)| = r|X| = r|Y|$ so that $|X| = |Y|$. Now, to show G has a perfect matching, one must show that for all $U \subseteq X$, $|N(U)| \geq |U|$. Well, assume for contradiction that there exists $U \subseteq X$ with $|N(U)| < |U|$. Then, consider the number of edges, k, between U and $N(U)$. Clearly, $k = r|U|$. Also, $k \leq r|N(U)|$. So, now one has $k = \frac{|U|}{r} \leq \frac{|N(U)|}{r}$ which implies that $|U| \leq |N(U)|$, a contradiction. So, Hall's condition is satisfied which means that G has a perfect matching.

So, any cubic bipartite graph G has a perfect matching $M \subseteq E(G)$. Let $G' = (V(G), E(G) - M)$. Now, G' is a 2-regular bipartite graph. It also has a perfect matching M'. Let $G'' = (V(G), E(G) - M - M')$. Now, G'' is a 1-regular bipartite graph which means that its edge set exactly forms a perfect matching M''. Now, for each edge $e \in M$ orient 1 unit of flow from e_1 to e_2 where $e = e_1e_2$ and e_1 is its endpoint is X and e_2 is its endpoint in Y. Likewise for $e \in M'$ and $e \in M''$. Now, for each vertex $v \in X$, v is incident to exactly one edge e, e', e'' in each of M, M', M'' respectively. The net flow at v is

$$\sum_{\epsilon \in \delta^+(v)} f(\epsilon) - \sum_{\epsilon \in \delta^-(v)} f(\epsilon) = (1+1+1) - 0 \equiv 0 \bmod 3.$$

Also, for $v \in Y$ v is incident to exactly one edge e, e', e'' in each of M, M', M'' respectively. The net flow at v is

$$\sum_{\epsilon \in \delta^+(v)} f(\epsilon) - \sum_{\epsilon \in \delta^-(v)} f(\epsilon) = 0 - (1+1+1) \equiv 0 \bmod 3$$

and we see that flow conservation is satisfied. Also, $f(\epsilon) \neq 0$ for all $\epsilon \in E(G)$ so this is a nowhere zero $\mathbb{Z}_3$ flow.

Now, say that a cubic graph G has a nowhere zero 3-flow. (Now we are speaking about 3-flows not $\mathbb{Z}_3$ flows). We wish to show that G is bipartite. Namely, we wish to show that G has no odd cycles. Clearly, G has some edges with flow value 2 (with some orientation not specified). Otherwise, if it were the case that $f(e) = 1$ for all $e \in E$. Then, for any vertex

$v \in V(G)$, 3 edges are incoming to v, 3 edges are outgoing from v, 2 edges are incoming and 1 edge is outgoing, or 1 edge is incoming and 2 edges are outgoing. If 3 edges are incoming with flow value 1, then net flow is $-3 \neq 0$. If 3 edges are outgoing with flow value 1, then net flow is $3 \neq 0$. If 2 are incoming and 1 outgoing then net flow $-1 \neq 0$. If 1 is incoming and 2 are outgoing then net flow $1 \neq 0$. So, for each $v \in V(G)$, there is some edge adjacent to v with flow value 2 (orientation not specified). In particular, there are two types of vertices: vertices v of type 1, meaning that v has two incoming edges with flow value 1 and 1 outgoing edge of flow value 2, or v of type 2, meaning that v has 1 incoming edge of flow value 2, and two outgoing edges of flow value 1. Next, one notes that no vertices of the same type can be adjacent. For otherwise, if two vertices v, w of type 1 are adjacent, then either they share an edge of flow value 2 or they share an edge of flow value 1. If they share an edge of flow value 2, then without loss of generality such an edge is oriented from v to w. However, then w has an incoming edge of flow value 2, a contradiction to w being type 1. If they share an edge of flow value 1, then without loss of generality, such an edge is oriented from v to w. Then, v has an outgoing edge of flow value 1, a contradiction to v being type 1. Similarly, by an analogous argument two vertices of type 2, cannot be adjacent. So, v, w adjacent implies that exactly one is of type 1 and the other is of type 2. So, assume for contradiction that G has an odd cycle. Then, there must exist two vertices on the cycle with the same type, a contradiction. Thus, G is bipartite.

Q6. Prove that a $0, \pm 1$ matrix which is minimally not totally unimodular (i.e. having all square subdeterminants in $\{0, \pm 1\}$ except for the matrix itself) has determinant ± 2.

Citation: "Theory of Linear and Integer Programming" by Alexander Schrijver page 270. We will assume that B is a minimally totally unimodular matrix (n by n) with $|\det(B)| > 3$ for contradiction. So, assume $|\det(B)| > 3$. Now, construct the matrix $C := [BI]$. Then, similar to what I did in question 1(a), (see my row reducing algorithm, the idea is the same), we can add or subtract rows from each other and multiply columns by -1 to get a matrix C' such that (1) C' is a $\{0, -1, +1\}$ matrix (using what I said in 1(a)), (2) C' has the standard basis vectors $e_1, e_2, \ldots, e_n$ as some of it's columns (we note that this is always possible as taking $C' = C$ e.g. doing no row reduction already results in a matrix that satisfies this condition) and (3) the number of unit basis vectors which appear in the first n columns of C' is maximized. Now, such a C' has the form

$$\left(\begin{array}{c|c|c} I_k & & 0 \\ \hline 0 & B' & I_{n-k} \end{array}\right)$$

where I_k is a possibly empty k by k identity matrix, B' is an n by n $\{0, 1, -1\}$ matrix, and I_{n-k} is an identity matrix. Now, since we performed row operations on C to get to C' (and then multiplied some columns by -1), one could also say that we performed row operations to the first n columns of C to get the first n columns of C' and then multiplied some columns of C' by -1. Now, since all row operations we performed changed the determinant of the square submatrix of C given by the first n columns by ± 1 and then multiplying some columns of the resulting square submatrix consisting of the left n columns of C' further multiplied the determinant by ± 1, one has that $\det(C'_{[1:n]} = \pm\det(C_{[1:n]})$ and by construction of C, one has that $C_{[1:n]} = B$, which gives that $\det(C_{[1:n]}) = \det(B)$ and recall that $|\det(B)| > 3$ or even more generally $\det(B) \notin \{1,-1\}$, which means that the first n columns of C' cannot be the identity matrix because then, $\det(C'_{[1:n]} = \det(I_n) = 1 = \pm\det(C_{[1:n]}) = \pm\det(C_{[1:n]}) \notin \{1,-1\}$, a contradiction. So, $k < n$. Now, by the fact that $\det(C'_{[1:n]} = \pm\det(C_{[1:n]}) \neq 0$, one has that $C'_{[1:n]}$ has full row rank, meaning that there exists some $i \in [k+1:n], j \in [n]$ such that $C'^{i}_{j} = \pm 1$. Now, without loss of generality, let $i = k+1$. The fact that the number of basis column vectors which appear in the first n columns of C' is maximized means that we cannot use row operations to zero out the entries below and above C'^{i}_{j} in column j while still maintaining a $\{0,1,-1\}$ matrix. This means that there is some row index i' such that one cannot zero out $C'^{i'}_{j}$ by adding or subtracting C'^{i}_{j} while still maintaining a $\{0,1,-1\}$ vector. Obviously, $C'^{i'}_{j}, C'^{i}_{j} \in \{0,1,-1\}$ means that we could do so unless doing so would cause some other entry in the sum or difference to be ≤ -2 or ≥ 2. Namely, either we could subtract row i from row i' to zero out column j of row i' but there's a conflict in some j'th column when doing so, or we could add row i to row i' to zero out column j of row i' but there's a conflict in some j'th column when doing so. In the former case, one has that there is a square submatrix

$$C'^{\{i\}\cup\{i'\}}_{\{j\}\cup\{j'\}} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

or in the latter case there is a square submatrix

$$C'^{\{i\}\cup\{i'\}}_{\{j\}\cup\{j'\}} = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}.$$

One then notes that either matrix has determinant -2, which in particular means that if one considers the columns which are basis vectors $S := \{e_k : k \in [n] \setminus \{i, i'\}\}$. Then, the set of columns $S \cup \{C'_j, C'_{j'}\}$ forms a submatrix of C' with determinant ± 2, which in turn means that the corresponding square submatrix of C also had determinant ± 2 (since our row operations only multiplied the determinant of any square submatrix of C by ± 1 and then multiplying some columns by -1 further multiplied any subdeterminant by ± 1). However, this then implies that B has a square submatrix with determinant ± 2. Why? Well either the square submatrix of C with determinant ± 2 was the first n columns of C in which case we get the result right away, or it contained some column C_j with $j \geq n+1$, in that case, such a column is a basis vector. Then, expanding along this basis vector (recursively or "inductively" as needed if more than one of the columns of this square submatrix have index $> n$) (note that there is at least one column of this square submatrix of C with column index $\leq n$, otherwise they are all basis vectors giving determinant 1) one gets a square submatrix of B with determinant ± 2, a contradiction to minimality and we are done.

Q7. In class, we argued that Edmonds' (cardinality matching) algorithm runs in $O(mn^2)$ time (where $m = |E|$ and $n = |V|$). You are asked to improve this to $O(n^3)$ by arguing that the time taken between two augmentations can be reduced to $O(n^2)$.

$U := \{v \in V(G)$ unmatched $\}$ One initializes a set of even vertices which is $S_{even} := \{v \in V(G) :$ there exists $x \in U$ and an m alternating path from u to v of even length$\}$. $S_{odd} := \{v \in V(G) :$ there exists $x \in U$ and an m alternating path from u to v of odd length and $v \notin S_{even}\}$. $S_{free} := V(G) \setminus (S_{even} \cup S_{odd})$. Now, just as in the usual Edmond's Blossom algorithm one grows a forest from the set of even vertices to their neighbors in $\bar{U}$, then following the matched edges adjacent to those neighbors, and so on growing alternating paths. Then, whenever there is an edge between two even vertices in the same component of F one knows that we have found a blossom BUT instead of our usual procedure, one replaces this entire blossom with a vertex b^* which is now adjacent to the matching edge leading up to it. One then continues the breadth first search to expand F and augments paths whenever possible shrinking blossoms first whenever found. If we ever augment a path containing a vertex b^* which corresponded to a blossom we know that we can "rotate" the blossom accordingly as shown in class to augment the desired path through part of the blossom (on one side). By shrinking these blossoms to single vertices, one saves the desired time factor.