CSE 202 Homework 4
Network Flow and NP problem sample solutions

1. **Row-Column sums**

   Solution

   This problem can be solved using a straightforward application of network flows. Consider an $n \times n$ matrix with column sums of c_i and row sums r_i and unknown elements $m_{i,j}$. Define a flow network in the following way: a source vertex s with edges to n verticies, v_i, with capacities, r_i. Similarly, define a sink vertex t with edges to n verticies u_i with capacities c_i. Then define n^2 verticies, $x_{i,j}$, with edges $(v_i, x_{i,j})$ and $(x_{i,j}, u_j)$ all of capacity k. An integral maximum flow on this network determines a solution to the matrix problem where the flow along $(x_{i,j}, v_i) = (x_{i,j}, u_j) = m_{i,j}$. (Note that since all capacities are integers the flow algorithms discussed in CLR will return integral flow values)

   Proof: First, consider a solution to the row-column sum problem. This is a valid flow since no capacities are exceeded: $\sum_i m_{i,j} = \sum_i f(x_{i,j}, v_i) = r_i$ which is the capacity of all (only one) inputs to v_i (similarly for the columns). Further, all $m_{i,j} < k$, and all flows $(v_i, x_{i,j}) < k$ and $(x_{i,j}, u_j) < k$, so these flows are not exceeded.

   Second, there exists a cut in the network, $G - s$ with value $\sum_i r_i$. Therefore, we have a valid flow and a cut of the same value and therefore each must be a max-flow/min-cut.

2. **Approximating MAX-CLIQUE; CLR 37-2, p. 984**

   Part (a) proof:

   Let $G = (V, E)$. First, we demonstrate a clique in $G^{(k)}$ of size l^k where l is the size of the maximum clique in G. Consider the set of verticies, $C' = \{(v_1, v_2, \ldots, v_k) : v_i \in C\}$, where C is a maximum clique in G. Each vertex in C' is adjacent to every other vertex since for $(v_1, \ldots, v_k)$ and $(u_1, \ldots, u_k)$ an edge exists if any $u_i \neq v_i$. The size of C' is l^k and forms a clique in $G^{(k)}$.

   We show that there is no larger clique inductively. First, since $G^{(1)} = G$, it is clear for $k = 1$. Now, assume it is true for all $i < k$. Consider a largest clique C^k in $G^{(k)}$. Let $C_a = \{(a, v_2, \ldots, v_k) | (a, v_2, \ldots, v_k) \in C^k\}$ for $a \in V$. By induction, $|C_a| \leq l^{k-1}$. For $a \neq b$, either $(a, b) \in E$ or for every $(a, v_2, ldots, v_k) \in C_a$ and every $(b, u_2, \ldots, u_k) \in C_b$, $(v_2, \ldots, v_k)$ and $(u_2, \ldots, u_k)$ are connected in $G^{(k-1)}$. If a and b are not connected in G, we will take each vertex of C_b and replace its first component with a. The resulting set of vertices is still a clique of $G^{(k-1)}$ of the same size since the projections of C_a and C_b onto their last $(k-1)$ components are disjoint. ea replace C_b are indepen first vertex in a tuple of size k, a. The tuple-verticies that begin with a must form a clique on the remaining $k-1$ elements (and there are l^{k-1} of those by the induction). Further, for some other prefix, b, if $(a, b) \in G$, then the verticies prefixed by BOTH a and b must form a clique on the remaining verticies. That is, they must share the cliques on $(k-1)$-tuples. Therefore, if the prefixes are drawn from m verticies, and the largest clique is l, then $m - l$ edges are missing and $m - l$ vertices

must share $(k-1)$-tuple cliques with another vertex giving us a maximum of $m-(m-l)\times l^{(k-1)} = l \times l^{(k-1)} = l^k$ verticies in the k-tuple clique.

Part (b)

Given an polynomial c-approximation for MAX-CLIQUE, we can generate a e (for any constant e) approximation by running it on graph $G^{(k)}$, where $e > c^{1/k}$ and then taking the k-th root of the answer returned. This remains polynomial time because the problem is only blown up by a polynomial factor, $k = lg(c)/lg(e)$ which is a constant.