CS 4820, Spring 2019	**Homework 7, Problem 1**
Name: Kevin Klaben NetID: kek228	

(1) *(10 points)*

(1a) *(3 points)* Let $G = (A \cup B, E)$ be an undirected p-regular bipartite graph, i.e., every vertex in $A \cup B$ has degree exactly p. Prove that G has a perfect matching.

(1b) *(7 points)* Latin rectangles are well studied objects in combinatorics, and various popular puzzles are based on Latin rectangles. For integers $m \leq n$, any $m \times n$ matrix, with each entry in the set $\{1, \ldots, n\}$, such that each integer appears at most once in each row and at most once in each column is called an (m, n)-Latin rectangle. An (n, n)-Latin rectangle is called an n-Latin square. Prove that any (m, n)-Latin rectangle can be extended to an n-Latin square. Given as input an (m, n)-Latin rectangle, design an efficient algorithm that extends the Latin rectangle to an n-Latin square.

(1a)

Claim: A graph $G = (A \cup B, E)$ which is an undirected p-regular bipartite graph has a perfect matching.

Proof: We take graph G to be made of two sets of vertices A and B such that it is a bipartite graph. This means that for each vertex in A it is not connected to any other vertex in A. As the Graph is also p-regular this means that each vertex is connected to p other vertices. As each vertex is connected to p other vertices and is not connected to any within it's own subset (A not connected to A, B not connected to B) we can say that each vertex in A is connected to exactly p vertices in B and vice versa. If we let $|A|$=n then we know that there are p*n edges leaving A going to B and for each vertex of B it has p connections, then $|B|$=n as well and thus $|A| = |B|$. By Hall's theorem, if for all subsets of A, X_A,|neighbors of X_A in B| >=$|X_A|$, then G has a matching that covers every node of X. Thus, we want to show that each subset of A has at least as many neighbors in B as it's own cardinality. This is shown as for each subset of A, we let the size of this subset be called m. For each node in this subset, there are p connections to set B. Thus, there are mp total edges to B for this subset. As every node in the graph is p-regular, the minimum number of nodes in B that can be connected to by these mp edges is m nodes as each of these nodes in B can be connected by a maximum of p edges from A. Thus, as the minimum number of nodes in B which are neighbors of this subset of A, is m and size of this subset of A is also m, then we can see that |neighbors of X_A in B| >=$|X_A|$, for all subsets of A. Thus, we can conclude by Hall's marriage theorem, that there exists a matching in G which covers every node in A. As was shown earlier that $|A| = |B|$ and each node in A is covered, then each node in B must be also a part of this matching as a single node in B cannot be connected to more than 1 node in A. Thus, as for all nodes in A are connected to a node in B and $|A| = |B|$, each node in G is matched with another node in G and thus, this is a perfect matching. Thus, the claim holds.

(1b)

Algorithm:

LATIN((m,n)-Matrix)

initialize a graph $G = (A \cup B, E)$ where E={}, A={a_1,...,a_n}, and B={b_1,...,b_n}

TrueTab[]=n x n matrix initialized to False for all entries
for each i in 0..m-1 **do**
  for each j in 0..n-1 **do**
    int k=Matrix[i][j]
    TT[k][j]=TRUE
  end for
end for

for each i in 0..n-1 **do**
  for each j in 0..n-1 **do**
    if TT[i][j]=FALSE **then**
    E=$E\cup$ Edge(a_{i+1} to b_{i+1}, weight = ∞)
    end if
  end for
end for
$A \cup B$=$A \cup B \cup S \cup T$
for each i in 1..n **do**
  E=$E\cup$ Edge(S to a_i, weight = 1) $\cup$ Edge(b_i to T, weight = 1)
end for
while Ford-Fulkerson(G)!= no flow **do**
  Edgeset=Ford-Fulkerson(G)
  Remove each selected edge that goes from A to B from G
  For each removed edge (a_i to b_j) insert i to the matrix row m+1 column j
end while

Runtime Analysis:
The first set of for-loops will take O(mn) time as it is nested n loops for each m thus O(mn). The second set will be O(n^2) as it is a loop which goes n times nested inside another loop that will take n time thus O(n^2) time. The third for-loop will take O(n) time. Finally, the while loop will operate a maximum of n times in the case that the matrix is empty initially. Each loop runs Ford-Fulkerson which will take O(mC) time and as the max flow is C=n O(mn). As there are n edges selected which must be removed and the removal is done by iterating over the n^2 edges. Thus, O(n^3) for the removal of these edges. Finally, removed edges are added to the matrix in O(n) time. Thus overall runtime for the While loop is O(n^4). Thus, the overall runtime as m<=n is O(n^4).

Proof of Correctness:

Subclaim: For every (m,n)-Latin rectangle can be extended to an (m+1,n)-Latin rectangle given m¡n.
Proof: As a proof of this claim, as in the algorithm we construct a Graph $G = (A \cup B, E)$. A={a_1,...,a_n} to represent the the integers of the matrix 1..n, and B={b_1,...,b_n} to represent the entries in each column 1..n in the new row. The edge set is constructed to represent all edges connecting integers a to column b which they don't currently occupy in the first m rows and thus can be placed in in the new row m+1. We want to show that for each subset of A, X, $|X|$ <= |Neighbors of nodes in X|. Each integer a in A is of degree n-m as it has already been in m different columns and thus n-m columns remain for it to be placed in. Similarly, each column

b has degree n-m as in the first m rows, it saw m different symbols and thus n-m remain that can be placed in column b. As was shown in part (1a), given a graph that is bipartite and p regular, a perfect matching exists. Thus, as our graph is bipartite and n-m-regular, we conclude that a perfect matching exists. As a perfect matching exists, edges going from integer a to column b mean that a can be placed in column b in matrix row m+1. Thus, we conclude that every (m,n)-latin rectangle can be extended to an (m+1,n)-latin rectangle.

Claim: Any (m,n)-Latin rectangle can be extended to an n-Latin square.
Proof: By the subclaim, the (m,n) latin rectangle can be extended to an (m+1, n) rectangle. This extension can be repeated for all instances where m<n. Once, m=n, we are at an n-latin square. Thus, for all (m,n) latin rectangles where m<n we can extend to an n-latin square. In the case where m=n initially we have already reached an n-latin square and thus the claim holds in this case as well. Thus, the overall claim holds.