Introduction to Algorithms	**Guide to HW 7**
CS 4820, Spring 2019	

(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.

Solution. **(1a)** We use Hall's condition to conclude that G has a perfect matching. Consider any non-empty subset $A' \subseteq A$, and let $\Gamma(A') \subseteq B$ denote the set of neighbors of A'. Since G is a p-regular graph, it follows that $p \cdot |A'|$ edges (i.e., all the edges with one end in the set A') have one end point in the set $\Gamma(A')$. Further, since every vertex in $\Gamma(A')$ has degree p, it follows that the number of edges that can have an end point in $\Gamma(A')$ is exactly $p|\Gamma(A')|$. Thus, it must be that $p|\Gamma(A')| \geq p|A'|$, and hence $|\Gamma(A')| \geq |A'|$. This shows that G satisfies Hall's condition, and hence must have a perfect matching.

(1b) We prove that any (m, n)-Latin rectangle can be extended to an n-Latin square in the following way: for $m < n$, we show to extend (m, n)-Latin rectangle to an $(m + 1, n)$-Latin rectangle using the the following procedure.

1. Build a bipartite graph G with n left nodes and n right nodes, and add edges as follows: Connect a node i on the left with a node j on the right, if the column i in the Latin rectangle does not contain the integer j.
2. Find a perfect matching M in G, or output 'Impossible' if no such matching exists.
3. Add an $(m + 1)$'th row to the Latin square, with the i'th entry set to j if node i on the left is matched to node j on the right in M.

Iterating the above procedure for $n - m$ times will give an n-Latin square or output 'Impossible'.

We prove that the above procedure always finds a perfect matching in G, and thus Step 2 never outputs 'Impossible'. Hence we can always extend a Latin rectangle to a Latin square. To see this, we claim that G is a $(n - m)$-regular graph. Consider any node i on the left. Since, there are exactly $n - m$ integers missing in the column i (in the (m, n)-Latin rectangle), it follows that all left nodes in G have degree exactly $n - m$. Now consider any right node j in G. It's degree is the exactly the number of columns which do not have the integer j as an entry. Since each row contains all integers in the set $\{1, \ldots, n\}$ exactly once, it follows that j appears exactly m times. Thus, the degree of j in the graph G will be $n - m$ (since there are n columns, and j must appear in m distinct columns). It follows that G is a $(n - m)$-regular graph, and hence by $(1a)$, it follows that G has a perfect matching. This concludes the proof of correctness.

The time taken to construct the graph G is $O(mn)$. Further, using Ford-Fulkerson, we can find a perfect matching in G in time $O(n^2(n - m))$. Finally, filling in a new row takes time $O(n)$. Since this procedure is repeated for $n - m$ times, the running time of our algorithm is $O(n^2(n - m)^2)$.

(2) *(10 points)* You are organizing a carnival for the town of Ithaca. There are various artists, arriving from all over the globe, to perform in various shows as part of the carnival. A total of n artists are participating from a total of m countries. Further, there are a total of r shows. The following are the constraints on organizing the carnival:

1. Artist ℓ, subject to her skills, is only capable of participating in a subset of the shows given by a list L_ℓ.

2. The i'th show requires exactly n_i artists to perform.

3. Each show can have at most one artist from each country.

4. So as to be fair to all artists, each artist can participate in at most r' shows.

As part of the input, you will receive integers $n, m, r, r', n_1, \ldots, n_r$ and the capability list $L_1, \ldots, L_n$ of the artists. Your task is to design an efficient algorithm to determine if you can organize the carnival subject to the above restrictions. If indeed it is possible, you have to output the allocation of artists to shows.

Solution. Construct a flow network with the following vertices and edges.

- source s, sink t
- vertex u_i for each artist, i
- vertex v_{jk} for each show, j, and country, k
- vertex w_j for each show, j
- edge (s, u_i) with capacity r' for each artist, i
- edge $(u_i, v_{j,k[i]})$ with capacity 1 for each artist, i and show $j \in L_i$. Here $k[i]$ denotes the country that i belongs to.
- edge (v_{jk}, w_j) with capacity 1 for all shows, j, and countries, k
- edge (w_j, t) with capacity n_j for all shows, j.

Compute an integer-valued maximum flow in this network. If it saturates every edge (w_j, t) then output an assignment of artists to shows by specifying that artist i belongs to each show j such that $f(u_i, v_{j,k[i]}) = 1$. If the maximum flow does not saturate every edge (w_j, t) then it is impossible to fill the shows.

Assume that m (the number of countries) is less than or equal to n (the number of artists); otherwise modify the flow network construction above to omit the vertices v_{jk} whenever k is a country that has no artists. The graph has $O(nr)$ edges, and its minimum cut capacity is bounded above by mr, which is the capacity of the cut separating $\{s\} \cup \{u_i\} \cup \{v_{jk}\}$ from $\{w_j\} \cup t$. Hence the $O(|E| \cdot C)$ running time bound for Ford-Fulkerson in this case translates to $O(nmr^2)$.

For the proof of correctness, first assume that the algorithm outputs an assignment of artists to shows. Then the flow conservation equation at node u_i says that the number of shows to which i is assigned equals $f(s, u_i)$, which is at most r' because of the capacity constraint of edge (s, u_i). Using flow conservation, and the fact that f is an integer-valued flow which saturates each edge (w_j, t), along with the fact that each incoming edge of w_j has capacity 1, we may conclude that each node w_j has exactly n_j incoming edges with flow value 1. Each of those edges is of the form (v_{jk}, w_j), and again using flow conservation and integrality we can deduce that for each show j there are exactly n_j distinct nodes of the form v_{jk} (corresponding to n_j distinct countries) such that each of the nodes receives one unit of flow on an edge of the form (u_i, v_{jk}). That flow unit must come from an artist, i, who belongs to country k and has $j \in L_i$, since otherwise (u_i, v_{jk}) would not be an edge of the graph. Summarizing these considerations, we can deduce that the show assignment derived from the flow satisfies the constraints $1, 2$ and 3 specified in the problem statement.

Conversely, suppose that there exists an assignment of artists to shows satisfying the problem constraints. Then for every artist i assigned to show j there is a path in the flow network of the form $s \to u_i \to v_{j,k[i]} \to w_j \to t$, and there is a flow of value 1 that sends one unit of flow along each edge of this path. Summing up those flows for each artist-to-show assignment, we obtain a flow that satisfies the capacity constraints on every edge, and saturates every edge into t, because no artist is assigned to more than r' shows, no two artists on show j belong to the same country, and each show j has n_j artists assigned to it.

(3) Extra Credit Question *(5 points)* Give an example of a bipartite graph, with countably infinite nodes on each side, which satisfies Hall's condition but does not have a perfect matching.

Solution. Consider a graph $G = (A \cup B, E)$, where $A = \{1, 2\} \times \mathbb{N}$, and $B = \{\mathbb{N}\}$. We define the edge set as follows:

- For all $i, j \in \mathbb{N}$ with $i \leq j$, add an edge between the node $(1, j) \in A$ and the node $i \in B$.
- For all $i, j \in \mathbb{N}$, there add an edge between $(2, j) \in A$ and $i \in B$.

It is easy to verify that the above graph satisfies Hall's condition as follows: consider any $A' \subset A$. If A' contains a node $(2, i)$ for some integer $i > 0$, then it follows that $\Gamma(A') = B$, completing the proof in this case. Now suppose, $A' \subset \{1\} \times \mathbb{N}$. In this case, it again follows from definition that if A' is finite, then $|\Gamma(A')| = |A'|$, and $\Gamma(A')$ is countably infinite in the case that the cardinality of A' is countably infinite.

Now, we prove that G does not have a perfect matching. Suppose if possible M is a perfect matching in G. Then the node $(2, 1) \in A$ is matched to some $k \in B$. Now consider the set of nodes $A' = \{(1, 1), (1, 2), \ldots, (1, k)\}$. From our construction, $\Gamma(A') \subseteq \{1, \ldots, k\}$. Since $k \in B$ is already matched under M (to $(2, 1) \in A$), the nodes in A' must be matched to nodes in $\{1, \ldots, k - 1\}$ under M. This is a contradiction since the cardinality of A' is k.