**COL 702**

# Homework IV

*Due on Nov. 14, 2021*

Justify your answers with proper reasonings/proofs.

1. Suppose you are given an array $M[1...n, 1...n]$ of numbers, which may be positive, negative, or zero, and which are not necessarily integers. Describe an algorithm to find the largest sum of elements in any rectangular subarray of the form $M[i...i', j...j']$ Your algorithm should run in $O(n^3)$ time.

2. For a sequence of $n$ days, you are given subsets $S_1, S_2, \ldots, S_n$ of $\{1, 2, \ldots, k\}$. Think of $S_i$ as the subset of people who are available for work on day $i$. You need to pick exactly one person from $S_i$ for each of the days $i = 1, \ldots, n$. For a person $j$, let $\Delta_j$ denote the $\sum_{i:j \in S_i} \frac{1}{|S_i|}$. This is the expected number of times $j$ would be picked if we pick a random person from $S_i$ on each of days $i = 1, \ldots, n$. A selection of persons, one from each set $S_i$, is said to be good if each person $j$ is picked at most $\lceil \Delta_j \rceil$ times. Show that such a selection is always possible, and give an efficient algorithm to find such a selection.

3. Let $(u, v)$ be a directed edge in arbitrary flow network $G$. Prove that if there is a minimum $(s, t)$-cut $(S, T)$ such that $u \in S$ and $v \in T$, then there is no minimum cut $(S', T')$ such that $u \in T', v \in S'$). Note that by definition of cut, $s \in S, t \in T$, and similarly $s \in S', t \in T'$.

4. Let $G$ be an undirected graph and $s$ and $t$ be two special vertices in it. Give an efficient algorithms to find the maximum number of node disjoint paths from $s$ to $t$ (a set of paths from $s$ to $t$ are said to be node-disjoint if no two of them share a vertex other than $s$ or $t$).

5. Let $G$ be an undirected graph. For a subset $S$ of vertices, let $e(S)$ denote the number of edges which have both the end-points in $S$. Given a rational number $\alpha$, we would like to find out if there us a subset $S$ of vertices such that $\frac{e(S)}{|S|} \geq \alpha$. Give an efficient algorithm to solve this problem.