1. Suppose we are given an array $A[1..m][1..n]$ of non-negative real numbers such that each row and column sum is an integer. We want to round A to an integer matrix, replacing each entry x in A with either $\lceil x \rceil$ or $\lfloor x \rfloor$ while maintaining the sum of entries in any row or column of A. For example,

$$\begin{pmatrix} 1.2 & 3.4 & 2.4 \\ 3.9 & 4.0 & 2.1 \\ 7.9 & 1.6 & 0.5 \end{pmatrix} \text{ rounds to } \begin{pmatrix} 1 & 4 & 2 \\ 4 & 4 & 2 \\ 8 & 1 & 1 \end{pmatrix}.$$

   Describe an algorithm that either outputs a feasible rounding scheme or outputs that there isn't one.

2. A group of n people $p_1, ..., p_n$ are trying to figure out a schedule over the next n nights $d_1, ..., d_n$ such that each person cooks exactly once. For each person p_i, there is a set of nights S_i that the person is not able to cook.

   A feasible dinner schedule is an assignment of each peorson to a different night such that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j, then $d_j \notin S_i$.

   (a) Describe a bipartite graph G such that g has a perfect matching if and only if there is a feasible dinner schedule for the group.

   (b) Suppose we already have at erroneous schedule in which two people p_i and p_j have been assigned to cook on the same day d_l, while no one has been assigned to d_k, and everyone else is assigned correctly. Describe an algorithm that, in $O(n^2)$ time, decides whether or not there exists a feasible dinner schedule, and outputs a feasible schedule if one exists.

3. In the min-cost flow problem, we are given a graph G and have to send exactly k units of flow from s to t, each edge e has a capacity $c(e)$ and a cost $w(e)$ associated with it, and we want to minimize the cost of the flow, where the cost of a flow f is $\sum_{e \in E(G)} w(e) f(e)$. (The flow f must satisfy conservation and capacity constraints.)

   Assume that you have a black box that can solve the min-cost flow problem in polynomial time.

   (a) Describe how to compute k edge disjoint paths from s and t such that the total cost of these paths is minimized.

   (b) Suppose G is bipartite. Describe an algorithm that decides if this graph has a perfect matching, and, if so, outputs the cheapest such matching.

   (c) Banana has just released a new version of their iFifi – the first electronic gizmo that is simultaneously dishwasher-safe and available in two colors: black and midnight.

   Banana has k distribution centers $C_1, ..., C_k$, and center C_i has t_i iFifis in stock. You need to plan the distribution of the iFifis to the Banana stores. You have a list of n stores $S_1, ..., S_n$, and for each one of them, there is a quota f_i of how many iFifis they need. For every distribution center C_i and store S_j, you know the distance between them in miles (rounded up to an integer).

Sending a single iFifi from a distribution center C_i to a store S_i costs d_{ij} dollars. Describe an algorithm that computes the minimum cost way to send all the required iFifis from the centers to the stores.

4. Let $G = (V, E)$ be an undirected graph. A *dominating set* $S \subset V$ is a subset of vertices such that every vertex $v \in V$ is either in S or adjacent to a vertex in S. The *minimum dominating set* is a dominating set of the smallest cardinality among all dominating sets in G.

   Reduce minimum dominating set and minimum set cover to one another, in both directions.

5. A subset S of vertices in an undirected graph G is called *triangle-free* if the induced graph G_S has no 3-cycles (aka triangles). That is, for every three vertices $u, v, w \in V$, at least one of the three edges uv, uw, vw is absent from G.

   Prove that finding the size of the largest triangle-free subset is NP-hard.