1. You are given an $n \times n$ matrix $A$ with real entries. You would like to round each of the entries $x$ in the array to either $\lfloor x \rfloor$ or $\lceil x \rceil$ such that the row and column sums don’t change. Give an efficient algorithm which either achieves this rounding or declares that no such rounding is possible.

2. [KT-Chapter7] You have a collection of $n$ software applications, $\{1, \ldots, n\}$, running on an old system; and now you would like to port some of these to the new system. If you move application $i$ to the new system, you expect a net (monetary) benefit of $b_i \geq 0$. The different software applications interact with one another; if applications $i$ and $j$ have extensive interaction, then the you will incur an expense if you move one of $i$ or $j$ to the new system but not both – let’s denote this expense by $x_{ij} \geq 0$. So if the situation were really this simple, you would just port all $n$ applications, achieving a total benefit of $\sum_i b_i$. Unfortunately, there’s a problem. Due to small but fundamental incompatibilities between the two systems, there’s no way to port application 1 to the new system; it will have to remain on the old system. Nevertheless, it might still pay off to port some of the other applications, accruing the associated benefit and incurring the expense of the interaction between applications on different systems. So this is the question: which of the remaining applications, if any, should be moved? Give a polynomial-time algorithm to find a set $S \subseteq \{2, \ldots, n\}$ for which the sum of the benefits minus the expenses of moving the applications in $S$ to the new system is maximized.

3. Suppose in a directed graph $G$, there are $k$ edge-disjoint paths from $s$ to $t$ and from $t$ to $u$. Are there $k$ edge disjoint paths from $s$ to $u$?

4. [KT-Chapter7] In sociology, one often studies a graph $G$ in which nodes represent people, and edges represent those who are friends with each other. Let’s assume for purposes of this question that friendship is symmetric, so we can consider an undirected graph. Now, suppose we want to study this graph $G$, looking for a close-knit group of people. One way to formalize this notion would be as follows. For a subset $S$ of nodes let $e(S)$ denote the number of edges in $S$, i.e., the number of edges that have both ends in $S$. We define the cohesiveness of $S$ as $e(S)/|S|$. A natural thing to search for would be the set $S$ of people achieving the maximum cohesiveness. Give a polynomial time algorithm that takes a rational number $\alpha$ and determines whether there exists a set $S$ with cohesiveness at least $\alpha$. Give a polynomial time algorithm to find a set $S$ of nodes with maximum cohesiveness.