As usual, when asked to give an algorithm to solve a problem, you should provide a description of the algorithm, justify its correctness, and analyze its running time.

Problem 1

You are trying to solve the following puzzle. You are given the sums for each row and column of an $n \times n$ matrix of integers in the range $1, \ldots, M$, and wish to reconstruct a matrix that is consistent. In other words, your input is $M, r_1, \ldots, r_n, c_1, \ldots, c_n$. Your output should be a matrix $a_{i,j}$ of integers between 1 and $M$ so that $\sum_j a_{i,j} = c_j$ for $1 \leq j \leq n$ and $\sum_j a_{i,j} = r_i$ for $1 \leq i \leq n$; if no such matrix exists, you should output “Impossible”. Give an efficient algorithm for this problem.

Problem 2 (KT 6.11)

Suppose you are consulting for a company that manufactures PC equipment and ships it to distributors all over the country. For each of the next $n$ weeks, they have a projected supply $s_i$ of equipment (measured in pounds), which has to be shipped by an air freight carrier. Each week’s supply can be carried by one of two air freight companies, $A$ or $B$.

- Company $A$ charges a fixed rate $r$ per pound (so it costs $r \cdot s_i$ to ship a week’s supply $s_i$)
- Company $B$ makes contracts for a fixed amount $c$ per week, independent of the weight. However, contracts with company $B$ must be made in blocks of four consecutive weeks at a time.

A schedule for the PC company is a choice of air freight company ($A$ or $B$) for each of the $n$ weeks, with the restriction that company $B$, whenever it is chosen, must be chosen for blocks of four contiguous weeks at a time. The cost of the schedule is the total amount paid to company $A$ and $B$, according to the description above.

Give a polynomial time algorithm that takes a sequence of supply values $s_1, \ldots, s_n$ and returns a schedule of minimum cost.

Problem 3 (KT 7.5)

Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

Let $G$ be an arbitrary flow network, with a source $s$, a sink $t$, and a positive integer capacity $c_e$ on every edge $e$; and let $(A, B)$ be a minimum $s-t$ cut with respect to these capacities $\{c_e : e \in E\}$. Now, suppose we add 1 to every capacity; then $(A, B)$ is still a minimum $s-t$ cut with respect to these new capacities $\{1 + c_e : e \in E\}$. 

Problem 4

You are taking a class with $k$ projects. You have $H$ hours to divide among the projects, and will spend an integer amount of time on each project. For every project $i$, you are given an array $G_i[0..H]$ so that, if you spend $h$ hours on project $i$, your grade for that project will be $G_i[h]$. Naturally, $G_i$ is increasing with $h$, and spending no time on an assignment gets 0 points, $0 = G_i[0] \leq G_i[1] \leq G_i[2] \leq \ldots G_i[H]$.

You need to allocate $H$ hours among projects $1, \ldots, k$, i.e., find non-negative integers $h_1, \ldots, h_k$ with $\sum_i h_i = H$ in order to maximize $\sum_i G_i[h_i]$. Give an efficient algorithm to find an optimal allocation.