1. (5pts) **Data Storage:** There are \( n \) digital images. Being images of the same object they are essentially similar. Let \( c_{i,j} \) denote the size need to store the difference between images \( i \) and \( j \). To save space, we can just store one image, and some of the differences. For example, given image 1 and the difference \( c_{1,2} \), we can reconstruct image 2. Similarly, if we have image 1 and \( c_{1,2} \) and \( c_{2,3} \), we can reconstruct both 2 and 3. We would like to store one image (say the one with fewest number of bytes), and some differences \( c_{i,j} \)’s such that every other image can be reconstructed. Which \( c_{i,j} \)’s should we pick to minimize the total storage space?

2. (5 pts) **Secret Agents:** An intelligence service has \( n \) agents in enemy territory. Each agent knows some of the other agents and can arrange to meet them. For each such possible meeting, say between agent \( i \) and agent \( j \), any message passed between these agents will fall into hostile hands with probability \( p_{ij} \). Agent 1 (who is the group leader) wants to transmit a confidential message among all the agents while minimizing the probability that the message is intercepted. How should the meetings between agents be arranged? 

Note that if \( S \) is the set of pairs of agents \((i,j)\) that communicate in your solution, then the probability that the message is intercepted is \( 1 - \prod_{(i,j) \in S} (1 - p_{ij}) \).

[Hint: Use spanning trees and logarithms.]

3. **Jobs with deadlines:** There are \( n \) jobs and a single machine where these jobs can be executed. Job \( i \) requires \( p_i \) units of processing, and can only be executed (possibly non-contiguously) during the time interval \([r_i, d_i]\). Let us assume that \( p_i, r_i \) and \( d_i \) are integers. So, for each job \( i \), we need to assign \( p_i \) units of processing among the \( d_i - r_i \) time slots \([r_i, r_i+1], [r_i+1, r_i+2], \ldots, [d_i - 1, d_i]\).

(a) (5 pts) Show that all jobs can be completed if and only if, for every time interval \([s, t]\)

\[
P(s, t) \leq t - s \quad \text{where} \quad P(s, t) = \sum_{j: r_j \leq s, d_j \leq t} p_j \tag{1}
\]

i.e. \( P(s, t) \) is the total processing of jobs whose interval \([r_i, d_i]\) lies completely in \([s, t]\).

(b) (3 pts) As there are potentially \( O(D^2) \) intervals \([s, t]\), where \( D = \max d_i \), testing condition (1) is inefficient for large \( D \). Can you modify the above test so that its running time only depends on \( n \), and not on the magnitude of \( p_i, r_i, d_i \)’s.

[Hint: One can get away by just checking (1) for some of the intervals \([s, t]\).]

4. **Picking disjoint subsets:** Let \( A \) be a finite set with \( m \) elements, and let \( A_1, \ldots, A_n \) be subsets of \( A \). Let \( b_1, \ldots, b_n \) be positive integers. We would like to pick subsets \( B_i \subseteq A_i \) such that \( |B_i| = b_i \) and the \( B_i \)'s are disjoint (i.e. \( B_i \cap B_j = \emptyset \) for all \( 1 \leq i < j \leq n \)).

(a) (5 pts) Show that this possible if and only if it holds that

\[
|\bigcup_{i \in I} A_i| \geq \sum_{i \in I} b_i
\]
for all subsets of indices $I \subseteq \{1, \ldots, n\}$.

(b) (2 pts) Design a polynomial time algorithm to find such $B_i$'s (if they exist).

5. (4 pts) **Lectures and Courses:** There are $n$ lecturers and $m$ courses. For each lecturer $i$ there is a subset $S_i$ of courses that she can teach. For each course $j$, a minimum of $c_j$ (different) lecturers must be assigned. For each lecturer $i$, we have a bound $a_i$ on the maximum number of courses that she can be assigned to. Design an algorithm to find an assignment of lectures to courses (if it exists) that satisfies all the requirements.

6. (3 pts) **Minimum mean cycle:** Let $G$ be an undirected graph with edge weights $c_e$. For a cycle $C$, its *mean* weight is the total weight of the edges of $C$ divided by the number of edges of $C$. Suppose we have an algorithm to detect a negative weight cycle in a graph (this can be done for example using the Bellman Ford shortest path algorithm). Design an algorithm, that given a threshold $t$, detects whether $G$ has a cycle with mean weight less than $t$ or not.

7. (12 pts) **Fire Evacuation:** There is a building with $n$ rooms, with up to $p_i$ people in room $i$. We want to know if the building can be evacuated within $T$ time steps in case of a fire.

Let us model this as a graph. Each location (room, stair, exit etc.) is associated with a vertex. An edge $(i, j)$ with capacity $c_{ij}$ indicates a link between locations $i$ and $j$ that $c_{ij}$ people can pass through at each time step. It takes $t_{ij}$ time units to cross edge $(i, j)$ (we assume that $t_{ij}$’s are integers). If $p_i$ is the number of people initially at location $i$, design an efficient algorithm to determine if everyone can be evacuated in $T$ time units?

[Hint: Make one copy of this graph for each time step $t = 0, 1, \ldots, T$. Put edges and capacities appropriately in this “time-layered” graph.]

8. (8 pts) **Distributed Computation on two processors:** There are $n$ jobs (processes) and two processors. Job $i$ runs in $a_i$ time on processor 1 and in $b_i$ time on processor 2. The jobs also communicate among each other. So, if two jobs $i$ and $j$ are assigned to different processors, this is not desirable and an overhead of $c_{ij}$ is incurred.

Find an assignment of jobs to machines that minimizes the total execution time plus the total overhead. So if $S$ is the set of jobs assigned to processor 1 and $T$ is the set of jobs assigned to processor 2, then the objective is

$$\sum_{i \in S} a_i + \sum_{j \in T} b_j + \sum_{(i,j): i \in S, j \in T} c_{ij}.$$  

[Hint: Formulate this as a min-cut problem on an undirected graph]

9. (8 pts) **Learning Skills:** There are a bunch of tasks $T_1, \ldots, T_m$ and a set of skills $\{1, \ldots, n\}$. Task $T_i$ is associated with a set of skills $S_i$, and it can be done only if one possess all the skills in $S_i$. Completing task $T_i$ fetches a profit of $p_i$, but learning a skill $j$ costs $c_j$. What set of skills should one learn to maximize the net benefit, i.e. find the set $S$ of skills that maximizes

$$\sum_{i: S_i \subseteq S} p_i - \sum_{j \in S} c_j.$$  

[Hint: Relate this to the project management problem that we saw in the class]