Problem 1.

(a) Let $G = (V, E)$ be a finite, undirected graph. Let $\ell$ be the collection of $E$ such that

$$A \in \ell \iff A \text{ is an acyclic subset of } E$$

Show that $(E, \ell)$ is a matroid.

**Hint:** It might be helpful to consider components.

(b) Let $w : E \to \mathbb{R}^+$ be a function that assigns a non-negative weight to each element of $E$. Let $\text{IsAcyclic}$ be a predicate that checks whether a subset of $E$ is acyclic or not in $O(T)$ time.

Design an efficient algorithm to find a maximum weight acyclic subset of $E$. Analyze its runtime and argue the correctness/optimality.

Solution.
Problem 2. Suppose you are given a set \( S = \{a_1, a_2, ..., a_n\} \) of tasks, where task \( a_i \) requires \( p_i \) units of processing time to complete, once it has started. You have access to a computer to run these tasks one at a time. Let \( c_i \) be the completion time of task \( a_i \), i.e. the time at which task \( a_i \) completes processing. Your goal is to minimize the average completion time:

\[
\frac{1}{n} \sum_{i=1}^{n} c_i
\]

For example, suppose there are two tasks, \( a_1 \) and \( a_2 \), with \( p_1 = 3 \) and \( p_2 = 5 \), and consider the schedule in which \( a_2 \) runs first, followed by \( a_1 \). Then, \( c_2 = 5 \), \( c_1 = 8 \), and the average completion time is 6.5.

(a) Give an algorithm that schedules the tasks to minimize the average completion time. Each task must run non-preemptively, that is, once task \( a_i \) is started, it must run continuously for \( p_i \) units of time. Prove that your algorithm minimizes the average completion time, and prove the running time of your algorithm.

(b) Suppose now that the tasks are not available at once. Each task has a release time \( r_i \) before which it is not available to be processed. Suppose also that we allow preemption, meaning a task can be suspended and restarted later.

For example, a task \( a_i \) with processing time \( p_i = 6 \) may start running at time 1 and be preempted at time 4. It can then resume at time 10 but be preempted at time 11 and finally resume at time 13 and complete at time 15. Task \( a_i \) has run for a total of 6 time units, but its running time has been divided into three pieces. We say that the completion time of \( a_i \) is 15.

Give an algorithm that schedules the tasks so as to minimize the average completion time in this new scenario. Prove that your algorithm minimizes the average completion time, and state the running time of your algorithm.

Solution. □