1 Some basic observations about WTAP
Let $\alpha > 1$. Prove the following statements about the Weighted Tree Augmentation Problem:
(a) If there is an $\alpha$-approximation algorithm for the special case of WTAP, where the given tree is a binary tree, then there is an $\alpha$-approximation for general WTAP.
(b) If there is an $\alpha$-approximation algorithm for the special case of WTAP, where the two endpoints of every link are leaves of the given tree, then there is an $\alpha$-approximation for general WTAP.

2 WTAP on trees of diameter 2
Prove that the special case of WTAP where the given tree $G$ has diameter 2 can be solved in polynomial time.
*Hint: Use that the minimum-weight perfect matching problem can be solved in polynomial time.*

3 TAP with a complete set of links
Consider the unweighted special case of WTAP where the link set $L$ is complete, i.e., we have $L = \binom{V}{2}$ and $w(\ell) = 1$ for every link $\ell \in L$. Prove that this special case is solvable exactly in polynomial time.

4 Shadow Completeness
Let $(G, L, w)$ be an instance of WTAP and let $k$ be the number of leaves of the tree $G$.
(a) Prove that there exists an optimal solution $F \subseteq L$ such that every edge of $G$ is covered by at most $k$ links.
(b) Prove that if the instance is shadow complete, there exists an optimal solution $F \subseteq L$ such that every edge of $G$ is covered by at most $\lfloor \frac{k}{2} \rfloor$ links.
*Remark: One can use this to prove that WTAP is solvable in polynomial time by a dynamic programming algorithm if the number of leaves of the given tree is upper bounded by a constant.*

5 The Decomposition Theorem for WTAP
Prove that the condition of the decomposition theorem that the paths $P_u$ with $u \in U$ are disjoint is necessary. More precisely, prove that for any $k \in \mathbb{N}$, there exists an instance $(G, L, w)$ of WTAP and a set $U \subseteq L$ of up-links such that the following holds. For any partition $\Pi$ of an
optimal solution OPT into $k$-thin components, we have
\[ \sum_{C \in \Pi} w(\text{Drop}_U(C)) < \frac{1}{2} \cdot w(U). \]

Hint: You can choose $G$ to be a tree of diameter 2.
Remark: The bound $\frac{1}{2} \cdot w(U)$ can be replaced by $\delta \cdot w(U)$ for an arbitrary $\delta > 0$.

6 Relative Greedy Algorithm for Set Cover

Consider the Weighted Set Cover problem, where an instance consists of a finite ground set $E$ and a family $S$ of subsets of $E$ with weights $w : S \rightarrow \mathbb{R}_{\geq 0}$. The task is to find a minimum weight subfamily $F \subseteq S$ that covers $E$, i.e., with $E = \bigcup_{S \in F} S$.

Let $p := \max_{S \in S} |S|$ be the maximum cardinality of a set in $S$. Prove that the following algorithm is a $(1 + \ln(p))$-approximation algorithm for Weighted Set Cover:

1. For $e \in E$, define $w_e := \min_{S \in S : e \in S} w(S)$.
2. Let $F := \emptyset$ and let $U := E$ be the set of elements uncovered by $F$.
3. While $F$ is not a feasible solution, do the following:
   - Choose a set $S \in S$ that minimizes $w(S) \cdot \sum_{e \in S \cap U} w_e$.
   - Add $S$ to $F$.
   - Replace $U$ by $U \setminus S$.
4. Return $F$.

7 Local Search Algorithm for Set Cover

Consider the Weighted Set Cover problem; see Exercise 6. We may assume w.l.o.g. that for every set $S \in S$ also all its subsets $R \subseteq S$ are present in $S$ and we have $w(R) \leq w(S)$.

Let $p := \max_{S \in S} |S|$ be the maximum cardinality of a set in $S$. Prove that the following algorithm returns a solution of cost at most $H_p$ times the cost of an optimum solution, where $H_j := \sum_{i=1}^{j} \frac{1}{j}$ is the $j$-th harmonic number.

1. Let $F$ be an arbitrary set cover solution, where w.l.o.g. $F$ is a partition of $E$.
2. Do the following as long as it decreases the potential function $\Phi(F) := \sum_{S \in F} H_{|S|} \cdot w(S)$.
   - For each set $T \in F$ define $\overline{w}(e) := \frac{1}{|T|} w(T)$ for all $e \in T$. // $F$ is a partition of $E$.
   - Choose $S \in S$ that minimizes $H_{|S|} \cdot w(S) - \sum_{e \in S} \overline{w}(e)$.
   - Replace every set $T \in F$ by $T \setminus S$. // Then $F$ is a partition of $E \setminus S$.
   - Add $S$ to $F$.
3. Return $F$.

Remark: One can turn this algorithm into a polynomial-time algorithm while losing only an arbitrary small $\varepsilon > 0$ in the approximation ratio by stopping the algorithm if the potential $\Phi(F)$ decreases only slightly in a single iteration.