# PIZZO
## Lista na 9 stycznia

**Zadanie 1.** Czy istnieje NP-zupełny problem, którego wejściem jest para $(s, t)$, i który jest w FTP zarówno, gdy parametrem jest $s$, jak i gdy parametrem jest $t$?

**Zadanie 2.** Czy problem Clique jest w XP?

**Zadanie 3.** Graf jest $r$-regularny, gdy każdy jego wierzchołek ma stopień $r$. Udowodnij, że dla dowolnego $r$ problem kliki dla grafów $r$-regularnych jest w FPT. Następnie pokaż, że problem jest FPT również wtedy, gdy $r$ jest częścią parametru (czyli parametrem jest para $(k, r)$).

**Zadanie 4.** Pokaż że problem „dla danej formuły w 2CNF i liczby $k$, czy istnieje wartościowanie spełniające co najwyżej $k$ klauzul" można rozwiązać w czasie $2^k n^{O(1)}$, a zatem że problem jest w FPT.

*Wskazówka: pozbądź się najpierw zmiennych, które występują tylko pozytywnie lub tylko negatywnie. Następnie każdy wybór wartościowania zmiennej zmniejsza $k$.*

**Zadanie 5.** Pokaż, że problem „dla danego spójnego grafu $G$ i liczby $k$, czy istnieje drzewo spinające zawierające co najmniej $k$ liści" można rozwiązać w czasie $4^k n^{O(1)}$, a zatem że problem jest w FPT. *Rozbudowana wskazówka jest na odwrocie.*


# PIZZO
## Lista na 9 stycznia

**Zadanie 1.** Czy istnieje NP-zupełny problem, którego wejściem jest para $(s, t)$, i który jest w FTP zarówno, gdy parametrem jest $s$, jak i gdy parametrem jest $t$?

**Zadanie 2.** Czy problem Clique jest w XP?

**Zadanie 3.** Graf jest $r$-regularny, gdy każdy jego wierzchołek ma stopień $r$. Udowodnij, że dla dowolnego $r$ problem kliki dla grafów $r$-regularnych jest w FPT. Następnie pokaż, że problem jest FPT również wtedy, gdy $r$ jest częścią parametru (czyli parametrem jest para $(k, r)$).

**Zadanie 4.** Pokaż że problem „dla danej formuły w 2CNF i liczby $k$, czy istnieje wartościowanie spełniające co najwyżej $k$ klauzul" można rozwiązać w czasie $2^k n^{O(1)}$, a zatem że problem jest w FPT.

*Wskazówka: pozbądź się najpierw zmiennych, które występują tylko pozytywnie lub tylko negatywnie. Następnie każdy wybór wartościowania zmiennej zmniejsza $k$.*

**Zadanie 5.** Pokaż, że problem „dla danego spójnego grafu $G$ i liczby $k$, czy istnieje drzewo spinające zawierające co najmniej $k$ liści" można rozwiązać w czasie $4^k n^{O(1)}$, a zatem że problem jest w FPT. *Rozbudowana wskazówka jest na odwrocie.*

**3.15** Start with the following observation: if $G$ is connected and has a subtree $T$, rooted at some vertex $r$, with at least $k$ leaves, then $G$ has also a spanning tree with at least $k$ leaves. In our algorithm, we start by guessing a non-leaf vertex $r$ of the tree in question, and consider all further trees as rooted in $r$.

At every node of the search tree we have a subtree $T$ of $G$ rooted at node $r$. Let $L(T)$ denote the set of leaves of $T$. Our objective in this branch is to find a subtree with $k$ leaves (if it exists) that is a supergraph of $T$.

First, observe that if there exists a leaf $v \in L(T)$ with at least two neighbors in $V(G) \setminus V(T)$, then we can branch on $v$. In one subcase, we consider $v$ being an internal vertex of the tree in question, adding all edges $vu$ for $u \in N(v) \setminus V(T)$ to the tree $T$. In the second subcase, we consider $v$ being a leaf vertex of the tree in question, deleting all edges $vu$ for $u \in N(v) \setminus V(T)$ from the graph $G$. Show that this branching is correct. Observe that in every branching step we either fix one vertex to be a leaf, or increase the number of leaves of $T$. Hence, this branching leads to at most $2^{2k}$ leaves of the search tree.

We are left with the case when all vertices $v \in L(T)$ have at most one neighbor outside $V(T)$. Prove that in this case a similar branching is still valid: for a vertex $v \in L(T)$, either proclaim $v$ a leaf, or fix as internal vertices both $v$ and all its descendants, up to the closest vertex of degree at least three, inclusive.

In the leaves of the search tree we have trees $T$ with $|L(T)| \geq k$ (where we report that $(G, k)$ is a yes-instance), or trees $T$ where all the leaves have been fixed as leaves, but still $|L(T)| < k$ (where we report no solution in this branch).

**3.15** Start with the following observation: if $G$ is connected and has a subtree $T$, rooted at some vertex $r$, with at least $k$ leaves, then $G$ has also a spanning tree with at least $k$ leaves. In our algorithm, we start by guessing a non-leaf vertex $r$ of the tree in question, and consider all further trees as rooted in $r$.

At every node of the search tree we have a subtree $T$ of $G$ rooted at node $r$. Let $L(T)$ denote the set of leaves of $T$. Our objective in this branch is to find a subtree with $k$ leaves (if it exists) that is a supergraph of $T$.

First, observe that if there exists a leaf $v \in L(T)$ with at least two neighbors in $V(G) \setminus V(T)$, then we can branch on $v$. In one subcase, we consider $v$ being an internal vertex of the tree in question, adding all edges $vu$ for $u \in N(v) \setminus V(T)$ to the tree $T$. In the second subcase, we consider $v$ being a leaf vertex of the tree in question, deleting all edges $vu$ for $u \in N(v) \setminus V(T)$ from the graph $G$. Show that this branching is correct. Observe that in every branching step we either fix one vertex to be a leaf, or increase the number of leaves of $T$. Hence, this branching leads to at most $2^{2k}$ leaves of the search tree.

We are left with the case when all vertices $v \in L(T)$ have at most one neighbor outside $V(T)$. Prove that in this case a similar branching is still valid: for a vertex $v \in L(T)$, either proclaim $v$ a leaf, or fix as internal vertices both $v$ and all its descendants, up to the closest vertex of degree at least three, inclusive.

In the leaves of the search tree we have trees $T$ with $|L(T)| \geq k$ (where we report that $(G, k)$ is a yes-instance), or trees $T$ where all the leaves have been fixed as leaves, but still $|L(T)| < k$ (where we report no solution in this branch).