TD 2: Trees

1 Degrees in Trees

For the following statements, decide if they are true or false and give a proof or counter-example.

1. There exists a tree where at least half the vertices have degree 1.
2. There exists a tree where at least half the vertices have degree 2.
3. There exists a tree where at least half the vertices have degree 3.
4. Suppose that in a tree T no vertex has degree 2. Then, at least half of its vertices have degree 1.
5. Suppose that a tree has a vertex of degree k. Then, the tree has at least k leaves.
6. Every tree with exactly two leaves is isomorphic to a path P_n.

Solution:

1. True. A star $K_{1,n}$.
2. True. A path P_n.
3. False. Let T be a tree on n vertices. Since T is connected, all vertices have degree at least 1. If at least $n/2$ vertices have degree 3, then $2m = \sum_v \deg(v) \geq \frac{3n}{2} + \frac{n}{2} = 2n$, but a tree must have at most $n-1$ edges.
4. True. Otherwise, we would have at least half the vertices with degree at least 3, which is not possible as in the previous question.
5. True. Obvious for $k \leq 2$. We proceed by induction and consider a tree T with a vertex v of degree $k+1$. Remove an edge e incident on v and we obtain two trees T_1, T_2, with $v \in T_1$. By induction, T_1 has at least k leaves, as v has degree k. T_2 has at least 2 leaves (as all trees do). Adding the edge e back only affects the degree of v (which is not a leaf in T_1, since $k \geq 3$) and at most one of the leaves of T_2, so the original tree has at least $k+1$ leaves.
6. True. If a tree is not isomorphic to P_n it must have a vertex of degree ≥ 3, so by the previous question would have more than 2 leaves.

2 Balanced Edge Separators

Recall that every tree has a balanced **vertex** separator, that is, a vertex whose removal leaves a collection of trees of size at most $\frac{n}{2}$. The question in this exercise is whether a similar claim can be made for **edge** separators.

1. Is it true that for all trees $T = (V, E)$ on n vertices there exists $e \in E$ such that both components of $T - e$ have at most $\frac{n}{2}$ vertices? How about $\frac{2n}{3}$ vertices? $\frac{3n}{4}$?
2. Show that in any tree T on n vertices of maximum degree Δ there exists an edge e such that both components of $T - e$ have at most $\frac{(\Delta-1)n+1}{\Delta}$ vertices.

Solution:

1. False. Example: a star $K_{1,n}$.

2. Take a balanced vertex separator v of T and let $T_1, \dots, T_k$ be the components of $T - v$ for $k \leq \Delta$. Let T_1 be the largest such component. We then set e to be the edge connecting v to T_1. We claim this edge satisfies the property as $T - e$ has two components: T_1 which (since v is a balanced separator) has size at most $\frac{n}{2} \leq \frac{(\Delta-1)n}{\Delta}$ (we assume that $\Delta \geq 2$ as otherwise the problem is trivial); and $T - T_1$ which has $1 + \sum_{i \in [2,k]} |T_i| = n - |T_1| \leq n - \frac{n-1}{\Delta} \leq n - \frac{n}{\Delta} + \frac{1}{\Delta} \leq \frac{(\Delta-1)n+1}{\Delta}$ vertices at most, where we used that T_1 is the largest of the trees, so $|T_1| \geq \frac{n-1}{\Delta}$.

   The bound is tight: in a $K_{1,\Delta}$ we have $n = \Delta + 1$, the maximum degree is Δ and removing any edge leaves a large component with Δ vertices, while the bound gives $\frac{(\Delta-1)(\Delta+1)+1}{\Delta} = \Delta$.

3 Polynomial-Time Algorithms

Give polynomial-time algorithms which take as input a tree T and calculate:

1. An independent set of T of maximum size
2. A clique of T of maximum size
3. The longest simple path contained in T
4. A dominating set of T of minimum size

Solution:

1. Greedy algorithm based on the following ideas:

   (a) If a graph G is disconnected and C is a connected component, $\alpha(G) = \alpha(C) + \alpha(G - C)$. (Here $\alpha(G)$ is the size of a maximum independent set). Proof: easy…

   (b) $\alpha(K_1) = 1$. Proof: trivial

   (c) (**Key idea**:) If in a graph G, v has degree 1, then $\alpha(G) = 1 + \alpha(G - N[v])$, where $N[v] = N(v) \cup \{v\}$ is the closed neighborhood of v. In other words, if a vertex has degree 1 it is always **safe** to take it in our set, but then we must remove all its neighbors from consideration. Proof: Clearly, $\alpha(G) \geq 1 + \alpha(G - N[v])$, because one possible independent set of G can be formed by taking the max independent set of $G - N[v]$ and adding v. Conversely, $\alpha(G - N[v]) \geq \alpha(G) - 1$ can be shown as follows. Take a maximum independent set S of G. If it contains no vertex of $N[v]$, then $S \cup \{v\}$ would also be independent, contradiction. If S contains one vertex of $N[v]$, then $|S \setminus N[v]| \geq \alpha(G) - 1$ and $S \setminus N[v]$ is an independent set of the new graph of the desired size. Finally, S cannot contain two vertices of $N[v]$, as $|N[v]| = 2$ and the two vertices of $N[v]$ are adjacent.

   Putting everything together we execute a simple greedy algorithm: as long as the graph contains edges, find a leaf, mark it as belonging in the independent set, and remove the leaf and its neighbor. The resulting set is a maximum independent set. Since finding a leaf can be done in polynomial time, this algorithm also runs in polynomial time.

2. Trivial: no tree can contain K_3 as a subgraph, as that would create a cycle. So, if the tree contains an edge, the maximum clique is K_2.

3. Key idea: for any two vertices x, y of T there is a unique path connecting them. Consider then all (n^2) pairs of vertices of T, calculate for each pair x, y the distance of an $x - y$ path, and find the largest value.

4. Consider a slightly more general problem: we are given a graph $G = (V, E)$ and a set $A \subseteq V$ of *optional* vertices. The goal is to find a minimum $S \subseteq V$ that dominates $V \setminus A$. In other words, the vertices of A are optional to dominate (but can be selected if desired). This is a generalization of DOMINATING SET, which is the special case where $A = \emptyset$. We give a recursive algorithm for this problem when G is a forest.

   The algorithm works as follows: if there is a non-optional leaf $v \in V \setminus A$ with neighbor u, we place u in the solution, set $A := A \cup \{N[u]\}$ and recurse; if there is an optional leaf $v \in A$ we delete v from the graph; if there is an isolated vertex v we take it in the solution if $v \notin A$, otherwise we delete it from the graph.

   The algorithm clearly works in polynomial time, because as long as the graph contains a leaf or an isolated vertex it will make progress and forests always contain such vertices. We now need to argue why the solution produced is optimal. Correctness is easy for vertices of degree 0, so we focus on leaves. If v is a non-optional leaf, we claim there is always an optimal solution S that contains the neighbor u of v. Indeed, any solution must contain u or v and if a solution contains v we can exchange it with u without increasing its size or leaving any vertex undominated. Since we know that u will be in the solution we can mark its neighbors as optional. If v is an optional leaf, there is always a solution that does not contain it, again using the argument that a solution can exchange v with its neighbor. Hence, deleting v is safe.

4 Edges and Cycles

Show that every connected graph G on n vertices and m edges contains at least $m - n + 1$ distinct cycles.
Solution:

We prove the statement by induction on m. For $m = 1$ we have $n = 2$ (since the graph is assumed connected), so $m - n + 1 = 0$ and the graph indeed contains at least 0 cycles. Similarly, for $m = 2$ the statement is vacuous, as we have $n = 3$, so $m - n + 1 = 0$. The first interesting case is $m = 3$ where we have either $n = 3$, so the graph is K_3 which indeed contains 1 cycle, or P_4, so $n = 4$, so $m - n + 1 = 0$.

Suppose now that the statement is proved for connected graphs with at most m edges and consider a connected graph G with $m + 1$ edges. We have two cases:

1. G is a tree. Therefore, $n = m + 2$ and the claim is that the graph has at least $(m + 1) - (m + 2) + 1 = 0$ cycles, which is correct.

2. G contains a cycle C. Let e be an edge of this cycle. Consider then the graph $G - e$, which has m edges and is connected, as we removed an edge of a cycle. $G - e$ has at least $m - n + 1$ distinct cycles. All these cycles are distinct from C, since C contains e (which is not in $G - e$), so G has at least $(m + 1) - n + 1$ distinct cycles, as desired.

5 Helly Property

Let T be a tree and $T_1, T_2, \ldots, T_k$ be k sub-trees of T. Suppose that for all $i, j \in [k]$ we have that $T_i \cap T_j \neq \emptyset$, that is, any two of the sub-trees share a vertex. Show that in this case there exists $v \in \cap_{i\in[k]} T_i$, that is, there exists a vertex that is common to all trees.
Solution:

Proof by induction: the claim is trivial for $k = 2$. Suppose that we have shown the statement for any collection of k sub-trees and we are given $k + 1$ sub-trees $T_1, \ldots, T_{k+1}$ such that any two share a vertex. By induction, the sub-trees $T_1, \ldots, T_k$ share a common vertex, call it x, and the sub-trees $T_2, \ldots, T_{k+1}$ share a common vertex, call it y. If $x = y$ we are done, so suppose $x \neq y$. Observe that all vertices in the path from x to y belong to $\cap_{i\in[2,k]} T_i$, so if one of these vertices belongs to $T_1 \cap T_{k+1}$ we are done. Suppose this is not the case and number the $x - y$ path as $x_1 = x, x_2, \ldots, x_r = y$. Consider the vertex x_i of this path which maximizes i but still has $x_i \in T_1$ and the vertex x_j of this path which minimizes j but still has $x_j \in T_{k+1}$. x_i, x_j are necessarily distinct and $i < j$. We now observe that $x_i, x_j \in T_1 \cup T_{k+1}$ and $T_1 \cup T_{k+1}$ is connected (as $T_1 \cap T_{k+1} \neq \emptyset$), so there is a path from x_i to x_j in $T_1 \cup T_{k+1}$ which does not use the edge $x_i x_{i+1}$. However, together with the path $x_i x_{i+1} \dots x_j$, this forms a cycle in T, contradiction.