P1) Let G be a connected undirected graph let T be the DFS tree with root s. Prove that for any edge $e = (u, v) \in G$, e is in a cycle in G iff one of the following holds:

- e is a non-tree edge,
- e is a tree edge (say u is parent of v) and there is a non-tree edge from a descendent of v to an ancestor of u.

Solution: In the previous section we discussed that any non-tree edge is in a cycle. In fact any non-tree edge together with the path between its endpoints in the tree makes a cycle.

Now, suppose $e = (u, v)$ is a tree edge and u is parent of v in T.

First, suppose there is a non-tree edge (x, y) such that x is an ancestor of u ad y is a descendent of v. Then, the path $u \to x \to y \to v \to u$ forms a cycle. In particular there is no repeated vertices because the path $u \to x$ goes over ancestors of u and the path $y \to v$ goes over descendents of v.

Conversely, suppose the tree edge (x, y) is in a cycle $y = v_0, \ldots, v_k = x, y$ for some $k \geq 2$ in G. Let S be the set of descendants of y in T (including y). Note that $x \notin S$ since x is the parent of y. Look at the smallest index in the cycle, say v_i, that does not belong to S. Such an index must exists since $v_0 \in S, v_k \notin S$. Then the edge (v_{i-1}, v_i) must be a non-tree edge. This is simply because there is only one tree edge out of S, the edge (x, y), and we know that $(x, y) \neq (v_{i-1}, v_i)$. Lastly, since every non-tree edge in the DFS tree is ancestor-descendant, and v_{i-1} is a descendant of y, v_i must an ancestor of x and y.

P2) Let G be a graph with n vertices such that the degree of every vertex of G is at most k. Prove that we can color vertices of G with $k+1$ colors such that the endpoints of every edge get two distinct colors.

Turn your proof into a polynomial time algorithm to color vertices of G with k colors.

Solution This problem is a bit more complex because there are two parameters that we can induct on: n and k. In this case, we let k be a **fixed** number in the entire proof and we will prove the statement by induction on n.

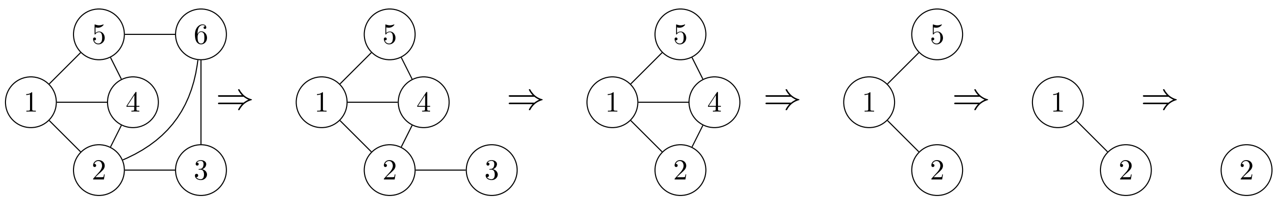
We prove by induction on n. First define $P(n)$ to be "every graph with n vertices such that the degree of every vertex is at most k can be colored with $k+1$ colors such that the endpoints of every edge have two distinct colors".

Base Case: $n = 1$. In this case we color the single vertex with a color. We can do so because $k \geq 0$.

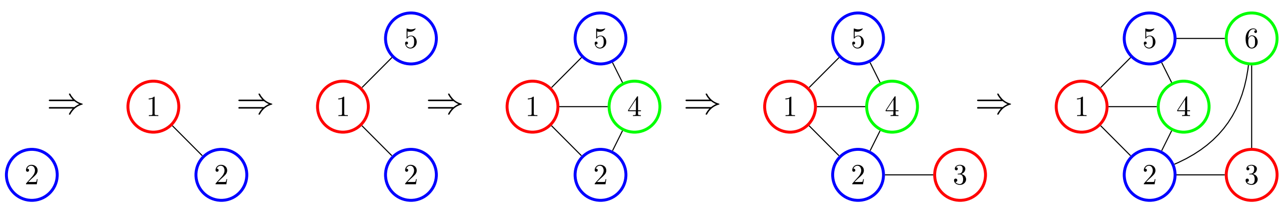
IH: Suppose $P(n-1)$ holds for some integer $n \geq 2$.

IS: We need to prove $P(n)$. Let G be an arbitrary graph with n vertices such that the degree of every vertex of G is at most k. Let v be an arbitrary vertex of G. Let $G' = G - v$ (we also remove all edges incident to v). Now, by removing v (and edges of v) we can only reduce degree of the rest of the vertices. Therefore, every vertex of G' also has degree at most k. Since G' has $n-1$ vertices by IH we can color vertices of G' with $k+1$ colors such that endpoints of every edge have distinct colors. Now, we color G. We color every vertex of G (except v) with the same color in G'. Now, to color v, note that it has at most k neighbors. Since we have $k+1$ colors there is a color that is not used in any of the neighbors of v. We color v with that color.

Algorithm: Note that this proof also gives an algorithm to color such a graph. Here is a sample execution of such an algorithm. Say $k = 3$, so we have 4 colors available. Say we remove vertices in the following order $6, 3, 4, 5, 1$.



Now, we can color. First, we color the last vertex 2 with blue. Then, we add back the removed vertices and each time we use a color not used on the neighbors: Note that to color the last



vertex 6 we got lucky. Even though it had 3 neighbors, two of them were color blue. So, we could color 6 with green and this way we used only 3 colors (of 4 available colors). We also had the option of coloring 6 with orange and that would also be a valid coloring.

Now, we write the algorithm to color vertices of G with colors $1, \ldots, k$.

```
Function Color(G,k)
    Initialize: Make all vertices uncolored
    for i = 1 → n do
        Let C[k+1] be an array of size k+1 initialized to False
        for j = 1 → i-1 do
            if j is colored a and j is a neighbor of i, Set C[a] = true;
        end
        Color i with any colors in C which is still false, i.e., unused.
    end
```

Algorithm 1: Algorithm for P3

P3) Prove or disprove: Every directed graph with n vertices and at least $n(n-1)/2+1$ directed edges has a cycle.

Solution: Since G does not have parallel edges the only possible way for G to have $> \binom{n}{2}$ edges is that there is a pair of vertices i, j such that both $i \to j, j \to i$ are edges of G. But then G has a cycle.