# 2192. All Ancestors of a Node in a Directed Acyclic Graph

## SOLUTION IN JAVA

class Solution {

public List<List<Integer>> getAncestors(int n, int[][] edges) {

List<List<Integer>> ans = new ArrayList<>();

List<Integer>[] graph = new List[n];

for (int i = 0; i < n; ++i) {

ans.add(new ArrayList<>());

graph[i] = new ArrayList<>();

}

for (int[] edge : edges) {

final int u = edge[0];

final int v = edge[1];

graph[u].add(v);

}

for (int i = 0; i < n; ++i)

dfs(graph, i, i, new boolean[n], ans);

return ans;

}

private void dfs(List<Integer>[] graph, int u, int ancestor, boolean[] seen,

List<List<Integer>> ans) {

seen[u] = true;

for (final int v : graph[u]) {

if (seen[v])

continue;

ans.get(v).add(ancestor);

dfs(graph, v, ancestor, seen, ans);

}

}

}