Step 4: Compute a layered layout

Reversing edges Your input is directed graph $D = (V, A)$ potentially with directed cycles. You need to find a *feedback set* - a subset of edges $A_f \subset A$ such that, when these edges are reversed, the resulting digraph, $D_f = (V, A \setminus A_f \cup \text{rev}(A_f))$ is acyclic. We have discussed two heuristics to achieve that, the trivial one and the one with guarantees by Eades, Lin, Smyth 1993. You can implement any of these two. Note that the reversed edges of the feedback set $\text{rev}(A_f)$ need to be in the digraph during the next stages of the framework, otherwise the visualization will be incorrect. After the final step of the framework, reverse the direction of $\text{rev}(A_f)$ to get the original edges A_f. Finally, please report the number of reversed edges for every graph you experiment with.

Visual choices Please display the layers explicitly as horizontal lines and not by only labeling the y-axis. Mark the reversed edges with a different color and/or texture. Display the arrows on the edges - this allows you to check whether the steps of reversing edges and layer assignments work correctly - after those you should not have edges pointing downwards.

Layer assignment For the layer assignment you are asked to perform height minimization, by computing so-called *topological numbering*. Please check your output layer assignment with example graphs and make sure that no edges are pointing downwards or are between the same layer. If something like this happens, there is a bug in your implementation.

Dummy vertices After the layer assignment check which edges connect non-consecutive layers. Each such edge has to be substituted by a path that has one intermediate vertex at each layer that the edge skips. These intermediate vertices need to be assigned to the corresponding intermediate layers. So, you need to augment your layer assignment with these newly created dummy vertices.

Crossing minimization You are asked to perform iterative crossing minimization using both median and barycenter heuristics. Let us call an *iteration* a single complete pass through all layers of a graph. Please report the number of crossings in a drawing after every iteration and explain how do you decide when to stop the process. You can consider plotting the number of crossings as a function of the number of iterations for every graph and every heuristic and using these plots discuss the power of the heuristics.

Complexity Analyse the worst-case computational complexity of every steps of your implementation.