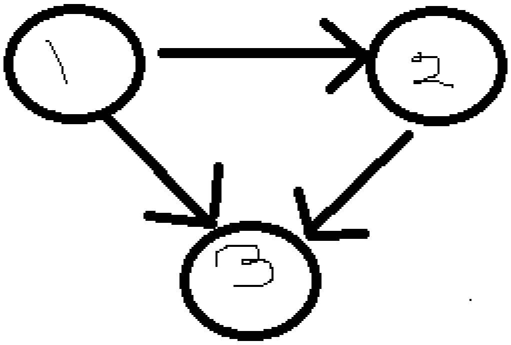
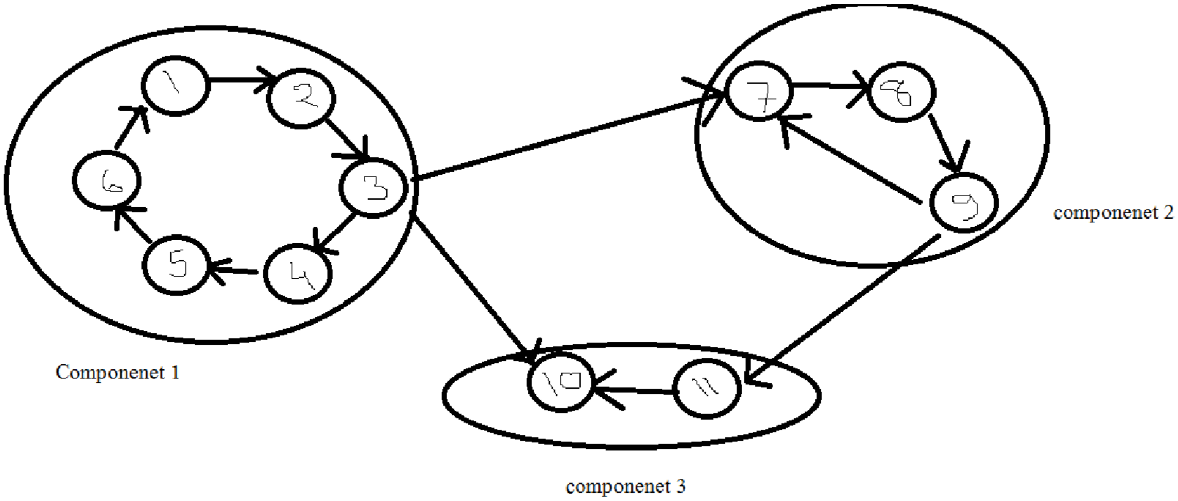


Look at this scenario. It says if we hit and miss 1st bird 2nd and 3rd bit will hear it. If we hit and miss 2nd bird 3rd bird will hear it. If we hit and miss third bird no one will hear because its outdegree is zero. Optimal solution here is to hit the third bird.



Look at the scenario. It's a bit complex one but when we have encircled the cycles as a single component doesn't it look like the first figure? Now out task is to find such a node in our new graph which out degree is zero. Because if we take such a node (A) which out degree is not zero it will obviously reach at such node which out degree is zero (B), then we why should take the nodes between the path A to B? We have to take minimum nodes because each node represent a bird. Therefore general solution is to build a graph using a stack . Do a SCC(Strongly Connected Component, tarjans algorithm) on that graph. And then, in the new graph, consider those nodes that has out degree zero and amongst them take the one that has minimum node in its component. Since SCC gives us a DAG there will be at least one such node that has out degree zero. Be careful about integer overflow.