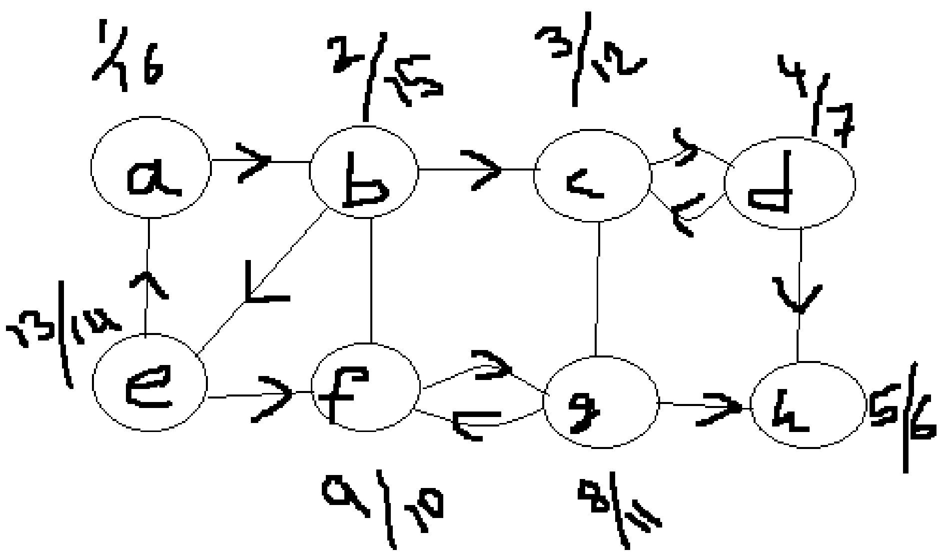
<u>Strongly Connected Components</u>

To find the strongly connected components from a given graph we need to know topological sorting. I have explained topological sorting in a previous lecture of mine. Finding strongly connected components involve 3 steps:

1. Topological sort
2. Draw the transpose of the graph
3. Run DFS on the transposed graph according to the order deduced in the topological sorting.

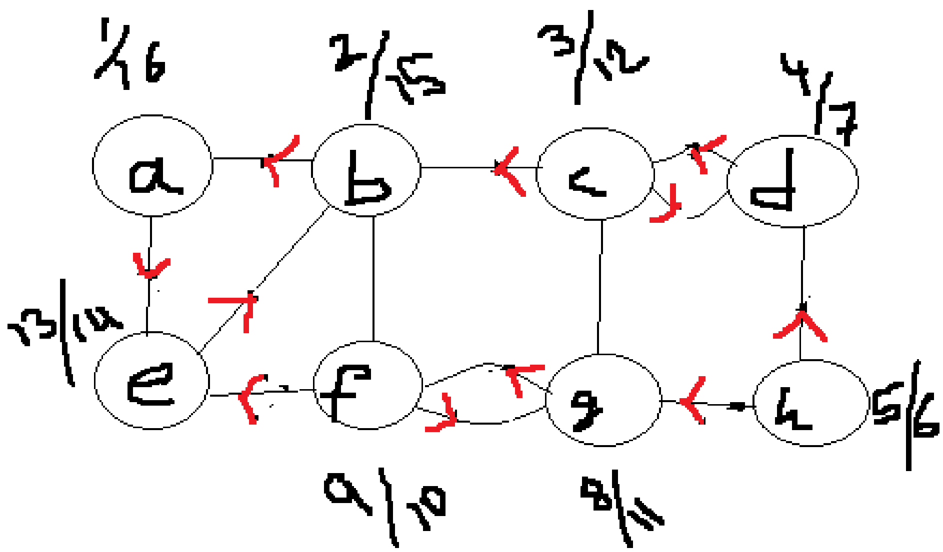
<u>Step 1</u>



Topological sort: **a, b, e, c, g, f, d, h** according to the ending time in decreasing order.

Step 2

The transpose of a graph is to the draw in the original graph **reversing** the direction of the edges.

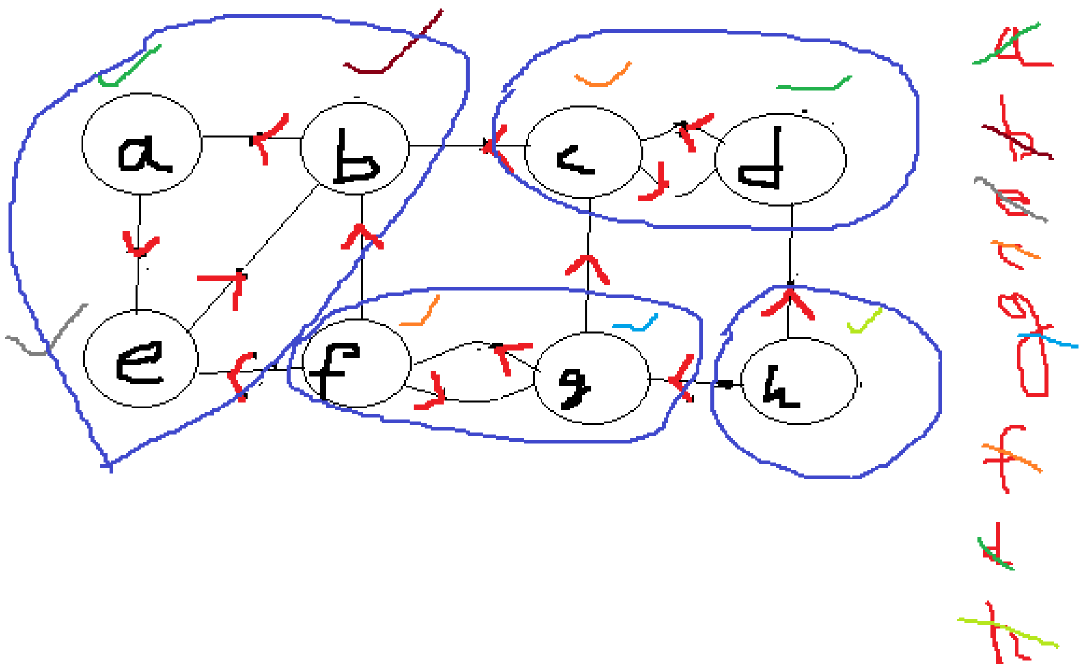


Step 3

We will run DFS again on this graph in the order found in the topological sort. The order was **a, b, e, c, g, f, d, h**. We will start from a. We don't need to record the starting and the ending time. The goal is to find the number of cycles.

We start from a them we go to its adjacent vertex e and from there to b. There are no adjacent vertex from b so we back track to e. E does not have any unvisited vertex. Therefore we can see that there is a cycle between a, e and b. Therefore these 3 vertices a, e and b belong to one component.

A, b and e has been visited and they do not have any more adjacent vertex. According to the topological order the next vertex to deal with is c. From c we go to d, d has one adjacent vertex which is c again. This is a cycle therefore c and d make another component. Now we start from g, and in the same way we go its adjacent vertex f. We cannot go anywhere from f other than g again, therefore these two form another cycle hence another connected component. The only remaining vertex is h which is itself a connected component because it does not have any unvisited vertex.



This is what the final picture looks like. This graph has 4 strongly connected components.