Homework#6 Graph traversal

Textbook:

9.16. Apply the strongly connected components algorithm on the directed graph shown in Fig. 9.10.

9.32. Design an efficient algorithm to determine whether a given graph is bipartite (see Sec. 3.3 for the definition of a bipartite graph).

Reverse graph. The *reverse* of a directed graph G = (V, E) is another directed graph GR = (V,E^R) on the same vertex set, but with all edges reversed; that is, E^R = {(v, u) : (u, v) ∈ E}.
Give a linear-time algorithm for computing the reverse of a graph in adjacency list format.

Find a cycle. Design a linear-time algorithm which, given an undirected graph G and a particular edge e in it, determines whether G has a cycle containing e.

Hamiltonian path in a DAG. Given a directed acyclic graph, design a linear-time algorithm to determine whether it has a Hamiltonian path (a simple path that visits every vertex), and if so, find one.