Introduction

As we have already seen, Dynamic programming can be applied on graphs to find the shortest paths using Bellman Ford and Floyd Warshall's Algorithm. In this chapter we will talk about some other applications of Dynamic programming on graphs.

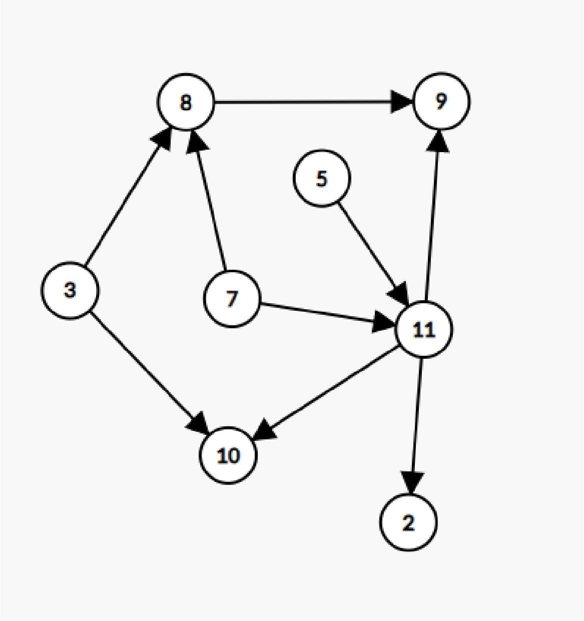
Important criteria for a problem to be solved by DP is the presence of a certain order of evaluation of DP states. That means that if the transitions from one state to another form a cycle then our DP will fail.

Since we will talk about graphs and generally we will keep nodes as a state of DP, it is safe to assume that the graph will be acyclic, i.e there will be no cycles in the graph. This will ensure that the graph has a topological order which will also be the order of evaluation of our DP.

What is a topological sort of a graph ?

As we have already discussed, topological sort is an ordering of vertices in a directed acyclic graph in which each node comes before all the nodes to which it has outgoing edges. As an example consider the course prerequisites structure at universities. A directed edge(v, w) indicates that course v must be completed before course w. Topological ordering for this example is the sequence that does not violate the prerequisite requirement.

Every directed acyclic graph must have one or more topological ordering. Topological Sort is not possible if the graph has a cycle since, for two vertices v and w in the cycle, v precedes w and w precedes v.



The topological sorts for above graph are: 7, 5, 3, 11, 8, 2, 9, 10
3, 5, 7, 8, 11, 2, 9, 10
and many more.