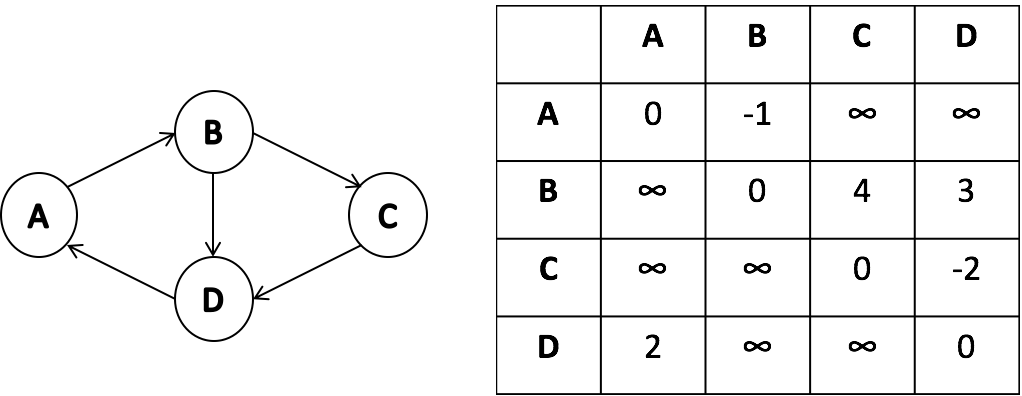
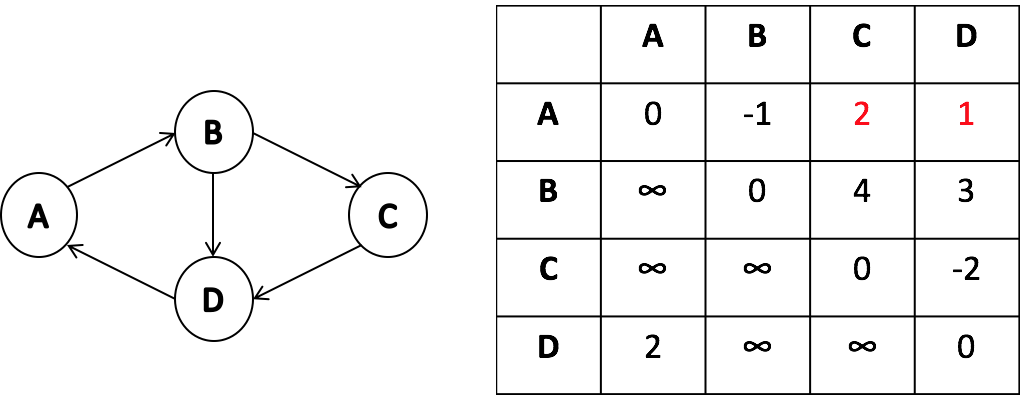
Floyd-Warshall’s Algorithm solves the problem Dijkstra's algorithm has with **negative** edges (not negative cycles, because those are mathematically undefined).

* Algorithm setup:
  + Maintain a table (matrix) that has the shortest known paths between vertices.
  + Initialize the table with three possible values:
    - self edges to 0
    - edges by edge weights
    - unknown paths to infinity



* Algorithm logic:
  + Consider adding every vertex to optimize the existing path:
    - vs.
    - vs.   
      * vs.



* + Now, do the same with the rest of the vertices (B, C, and D). At the end of the algorithm, we will have shortest paths for all pairs.
* Running time:
  + With Dijkstra’s algorithm we assumed optimality → once we find a path from A to B we do not try to find another path from A to B with shorter distance.
  + On the other hand Floyd-Warshall’s algorithm explores all possible paths to determine the shortest path. If we explored all possible paths with Dijkstra’s algorithm, the running time would have been much worse than .