Consider the graph G given.

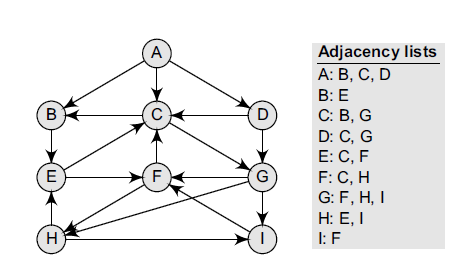
The adjacency list of G is also given. Assume that G

represents the daily flights between different cities and we

want to fly from city A to I with minimum stops. That is,

find the minimum path P from A to I given that every edge

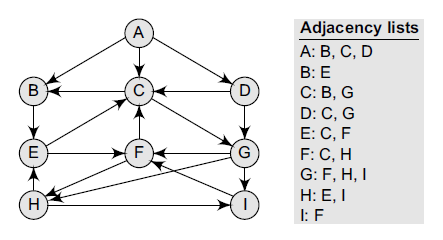
has a length of 1. Using BFS.



Consider the graph G given. The adjacency list of G is also given.

Suppose we want to print all the nodes that can be reached from the node H (including H itself).

use a depth-first search of G starting at node H.



stack: h

Top=1

1) process top node: h

Add all the adjacent of top node

Stack: E , I

TOP=2

2) PROCES TOP NODE: I

ADD ALL ADJACENT STACK: E ,F

TOP=2

3) PROCESS F

STACK: E , C

TOP =2

4) PROCESS C

STACK: E, B, G

TOP=3

5) PROCESS G

STACK: E,B ,TOP:2

6) PROCESS B

STACK: E TOP:1

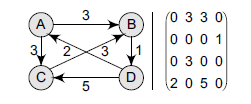
7) PROCESS E

STACK: ,TOP=0

8) STOP NOW SINCE STACK IS EMPTY.

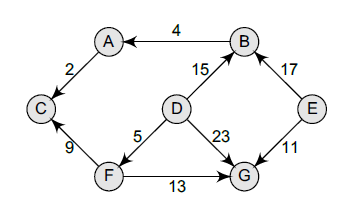
Consider a weighted graph G given and apply Warshall’s shortest

path algorithm to it.



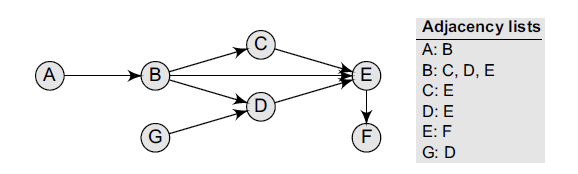
Consider the graph G given in Taking D as the initial node, execute

the Dijkstra’s algorithm on it.



Consider a directed acyclic graph G given.

find a topological sort T of G.



Q: A,G,B,C,D,E,F ONE ANSWER OF TOPOLOGICAL SORT

Q: G,A,B,C,D,E,F

## Dijkastra’s algo to find min. distance between single source to all nodes.

