chang@umbc.edu

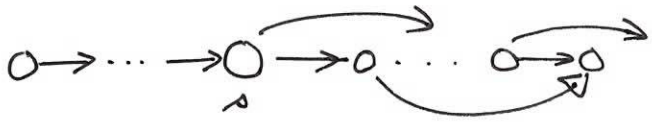
Shortest Path in a DAG

directed acyclic graph

no cycles ⇒ no negative cost cycles

Idea:

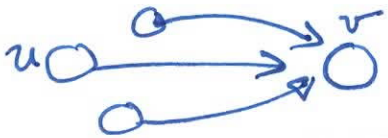
① Use topological sort



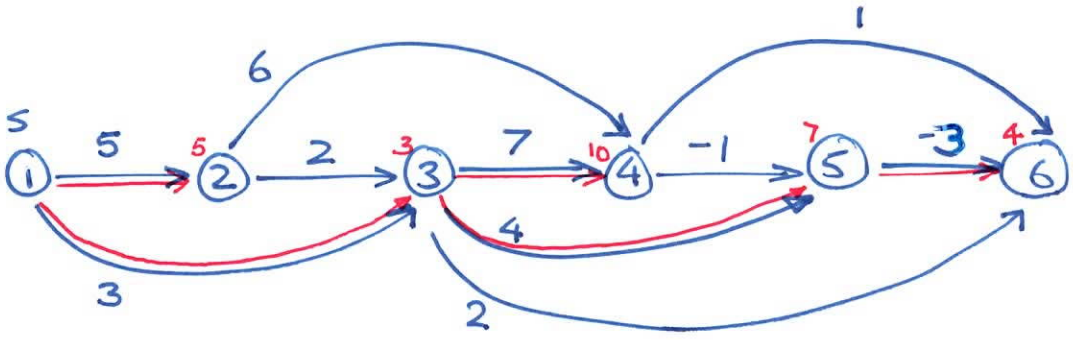
② All edges go left to right.

③ All vertices left of source vertex s are unreachable.

④ For each vertex v, in topological order, check every incoming edge (u,v) and pick u s.t. $dist[u] + w(u,v)$ is smallest



Example:



dist[]:

1	0
2	5
3	3
4	10
5	7
6	4

pred[]

1	nil
2	1
3	1
4	3
5	3
6	5

DAG_SHORTEST_PATH (G, s)

Topological_Sort (G)
for each $v \in V$, make dist[v] = ∞ & pred[v] = nil

dist[s] = 0, pred[s] = nil

for each $v \in V$ in topological order
    for each $(u,v) \in E$
        if dist[v] > dist[u] + w(u,v) then
            dist[v] = dist[u] + w(u,v)
            pred[v] = u

Running Time

$O(V+E)$ for Topological Sort

$O(V+E)$ to examine every edge of every vertex.

Total = $O(V+E)$