Date: 02/06/21

Black Board

Design and Analysis of Algorithms

Topics:

- Bellman Ford Algorithm ✓
  - Dynamic Programming for SSSPs

Single Sort Shortest Paths for various types of graphs

Directed or Undirected Un-weighted Graph	BFS	O(|V|+|E|)
Directed Acyclic Graph	DP based on Topological Sort	O(|V|+|E|)
Directed or Undirected Weighted Graph (+ve weights)	Greedy Algorithm (Dijkstra's Algorithm)	O((|V|+|E|)lg|V|)
Directed or Undirected Weighted Graph (+ve or –ve weights)	DP Algorithm (Bellman-Ford Algorithm)	O(|V||E|)

The main idea of the Bellman Ford Algorithm

The Relax Method

Relax(x, y, w)

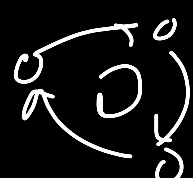
IF $dist[y] > dist[x] + w(x,y)$

$\{$ $dist[y] = dist[x] + w(x,y)$

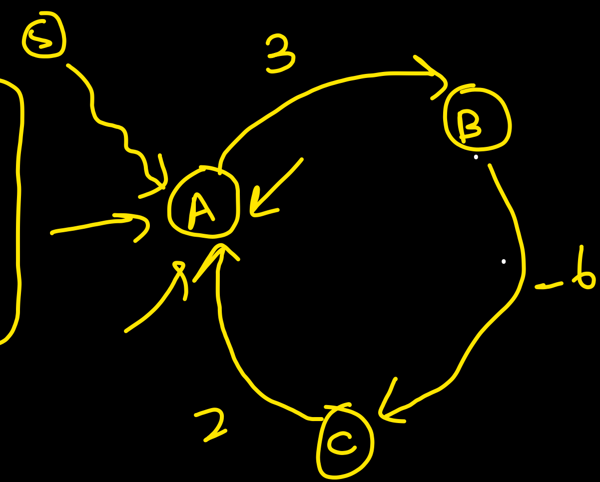
$par[y] = x$

return true

return false



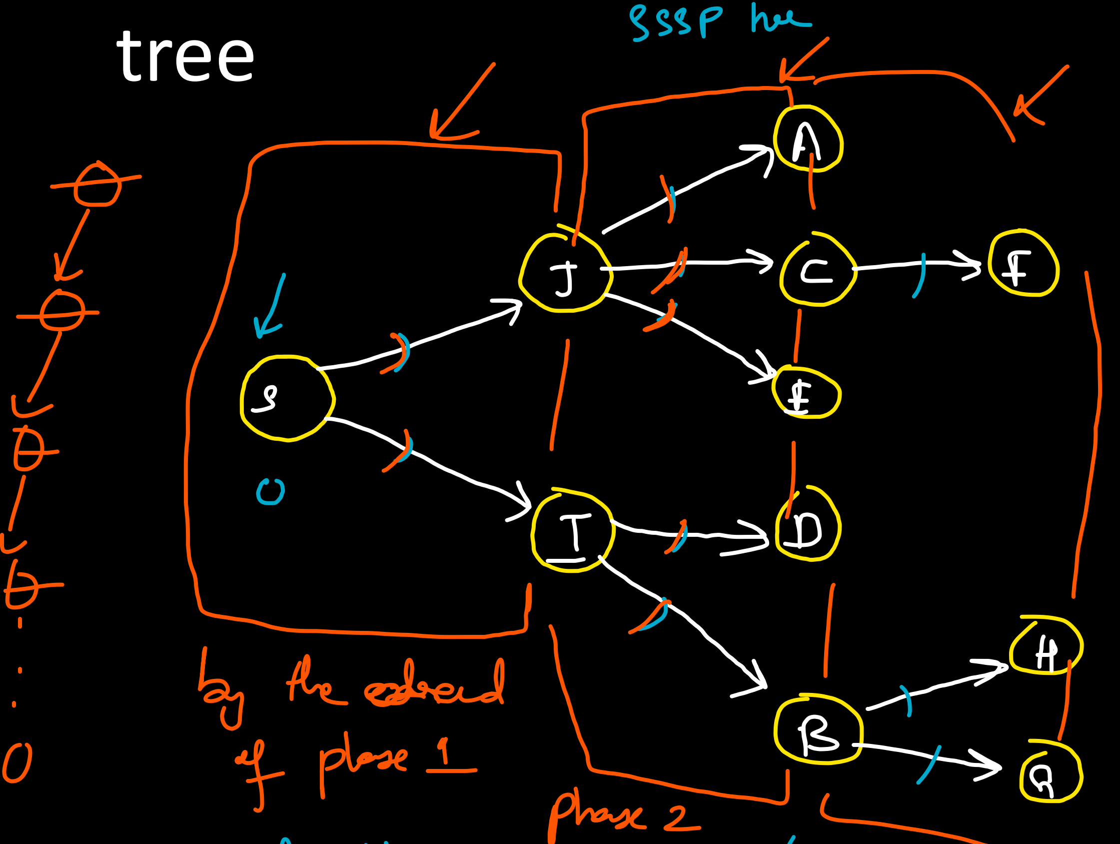
Negative cycle



Can be detected using Bellman Ford. How?

By adding a $(|V|+1)^{st}$ phase & seeing if a relax call returns true.

How to use the Relax Method to find the SSSP tree



- we are searching for the SSSP tree which is 'hiding' in the graph.
- Miraculously

relax(S, J, w)
relax(S, T, w)
⋮

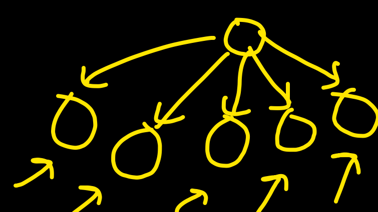
→ we don't know the "miraculous" order of relaxing the edges

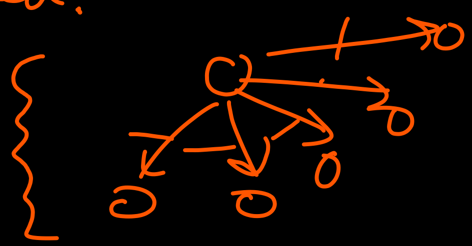
→ we do not know the correct order of relaxing the edges — therefore we need relax all edges in any order (repeatedly)

Phase: relax all edges in any order.

→ Every phase correctly (optimally) marks of nodes in the next level.

→ How many phases are needed?

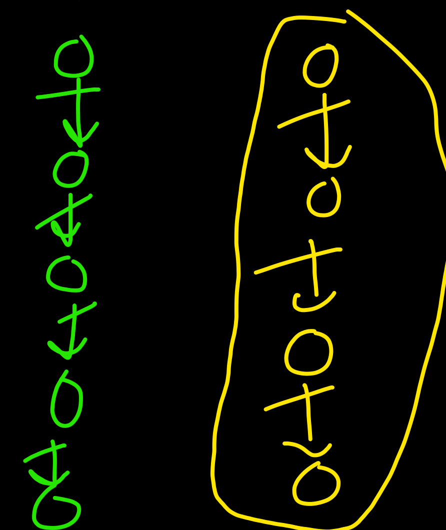




~~Bellman Ford Example~~

$T(G=(V,E)) = O(|V||E|)$

BellmanFord$(G=(V,E,w), s)$



For each $x \in E$:

$\quad dist(x) = \infty$

$\quad Par[x] = nil$

$dist[s] = 0$, flag = true, $i = 1$

WHILE $i < |V|$ AND flag == true:

flg = false

For each $(x,y) \in E$: // phase

flg = flg || Relax(x, y, w)

$i = i+1$

1 extra phase