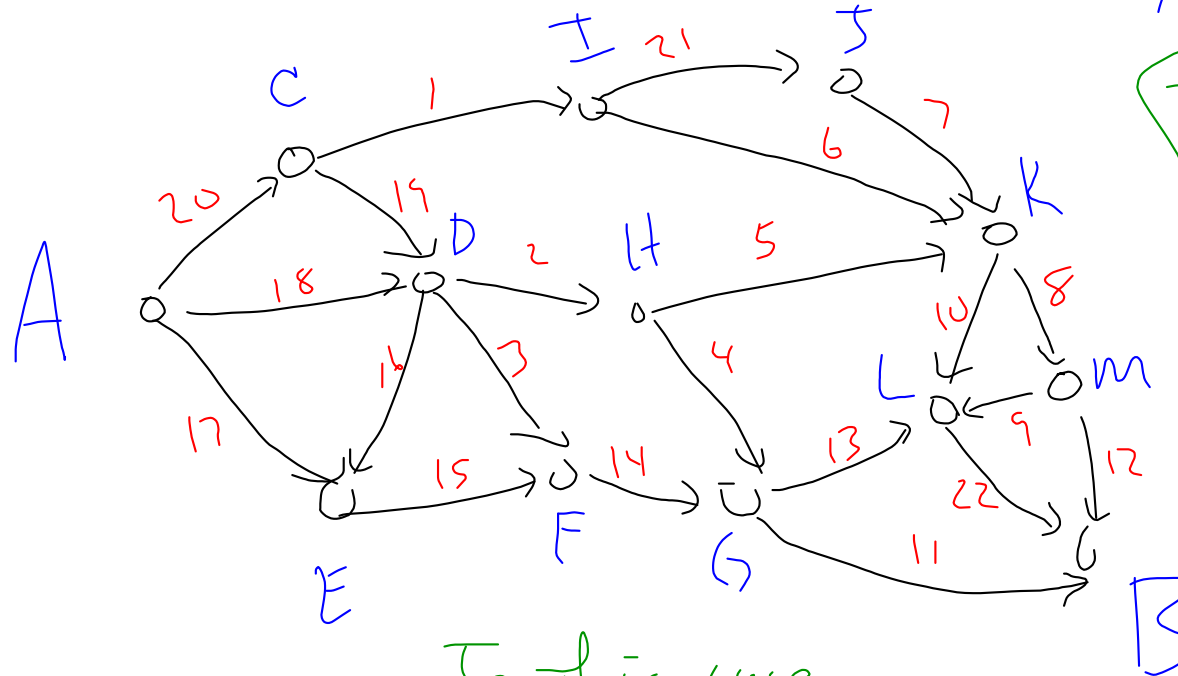


H!!! Last Day!

)) (??)

Dijkstra on a DAG.

Shortest path from A to B?



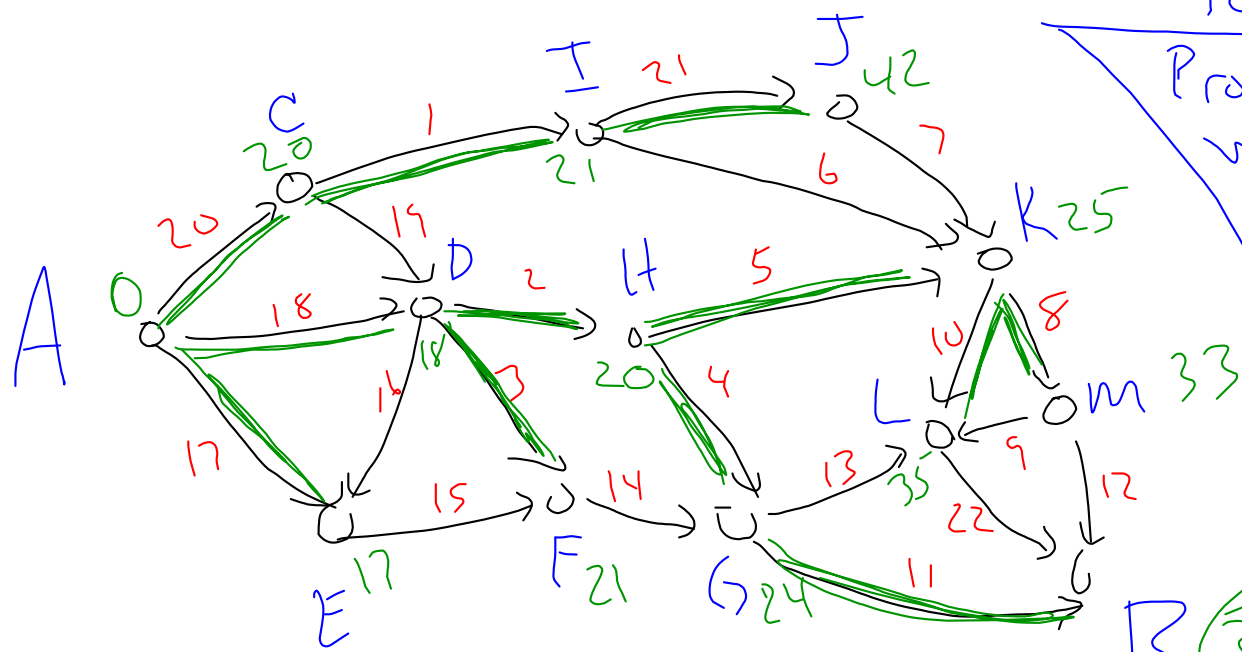
TO G - 28
 L - 17
 M - 21

Suppose there

In this case, Go through M.

were lengths of shortest paths from A

Dyn Prog Solution: Work away from A, solving the "smaller" problem (closer to A) before the larger problems. F.e. work in a ~~BFS~~ fashion.



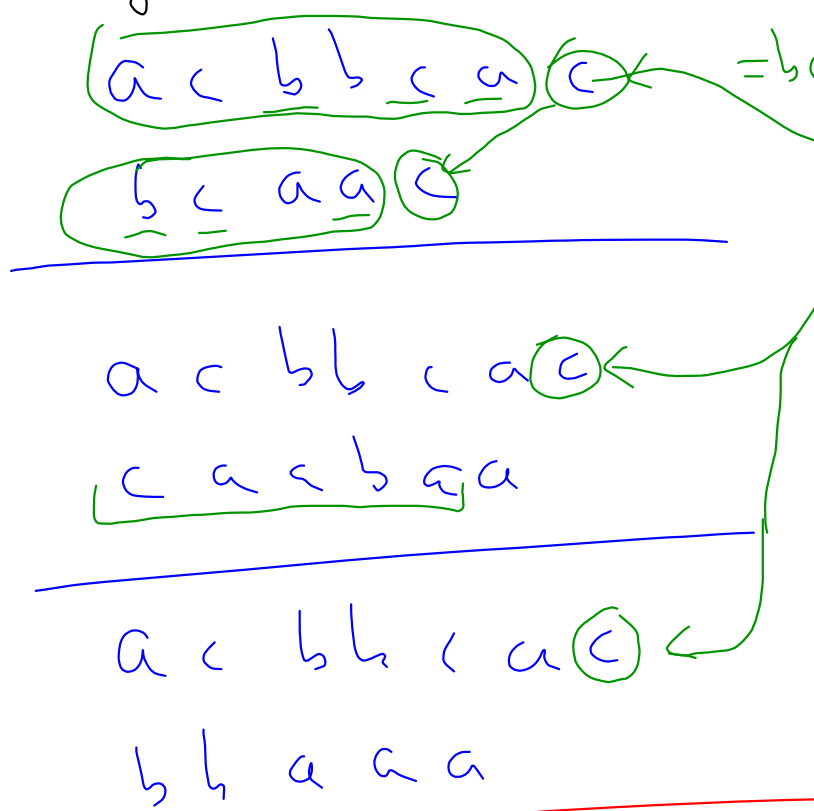
Top Sort
Process the vertices according to any topological sort.

Green memos = distances

$$d[v] = \min_{\text{predecessors } p} [d[p] + w(p,v)]$$

↑
weight on edge

Longest Common Substring (LCS) Smaller Problems

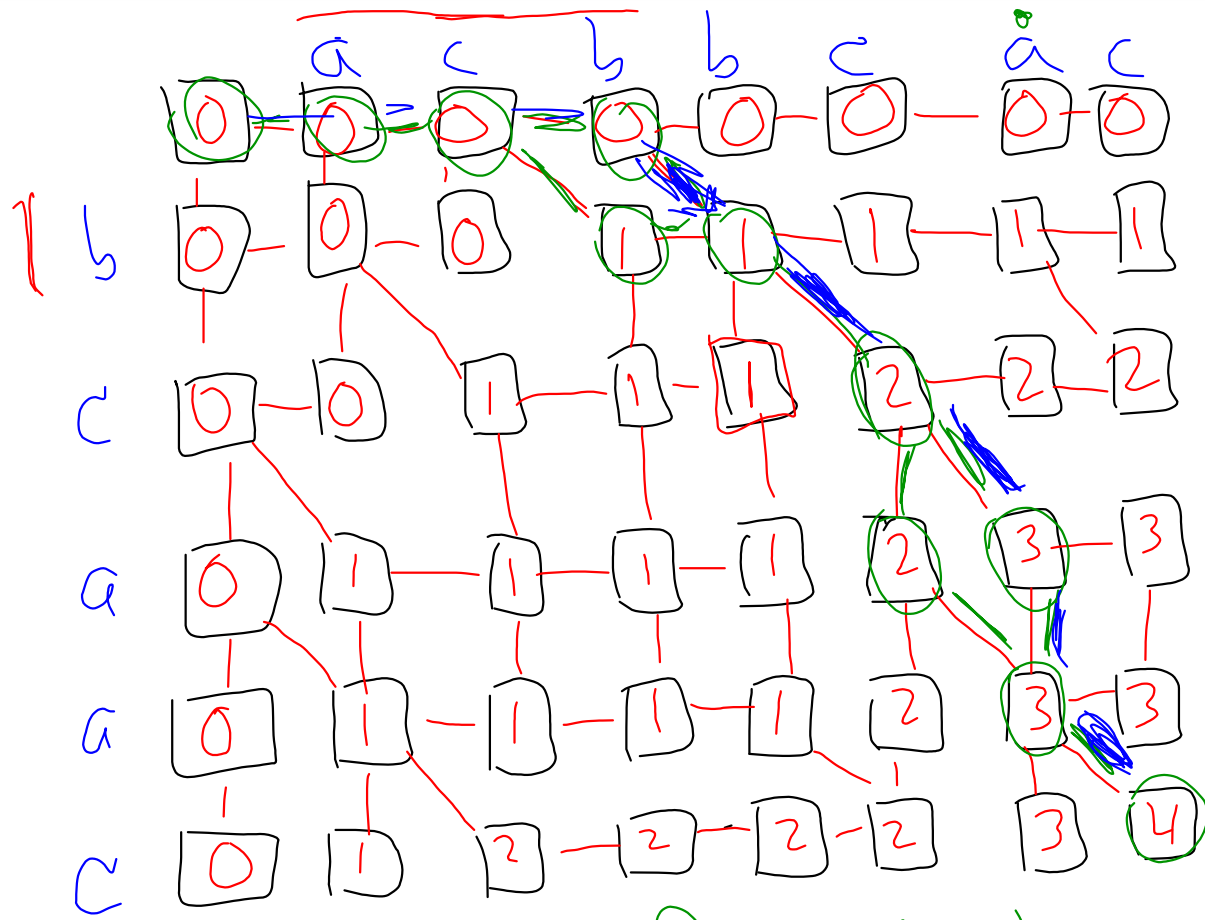


=bcac Q: Does the LCS contain this (c)?

A: If so, what does that c match with? If not, then our LCS is found by lopping off that letter and solving the resulting subproblem.

Recursion: $LCS(\text{string1}, \text{string2})$
 = longest of $\begin{cases} LCS(S, T) + "x" & \text{if } x = y \\ LCS(Sx, T) & \text{if } x \neq y \\ LCS(S, Ty) & \text{if } x \neq y \end{cases}$

String1 = (S) (X) single characters
 String2 = (T) (Y) characters
 ↖ strings



In each box goes the length of the LCS of the prefixes "to that point"

Pick any path back, write down diagonal steps

bcac

To find the LCS, start @ the end + walk back

The end