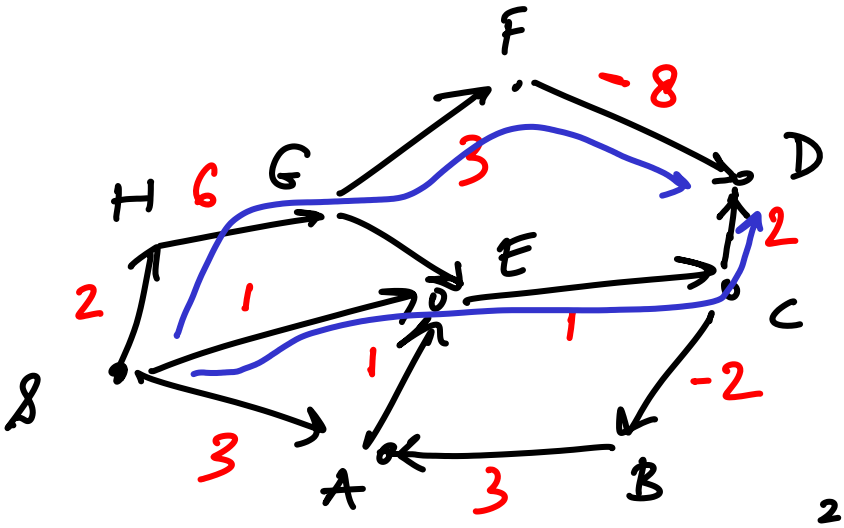
CSL 356 Lecture 39 Nov 12

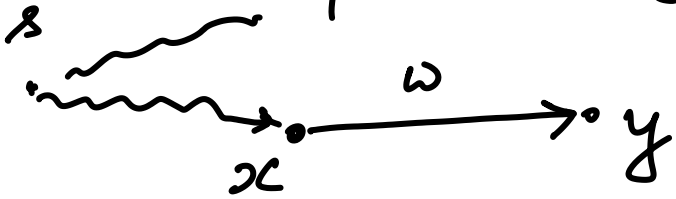
Shortest paths in weighted directed graphs



no -ve cycles

1. Find shortest path between $u \rightsquigarrow v$
2. Find shortest paths from s to all other vertices (single source shortest path)
3. Find shortest paths between all pairs of vertices. APSP

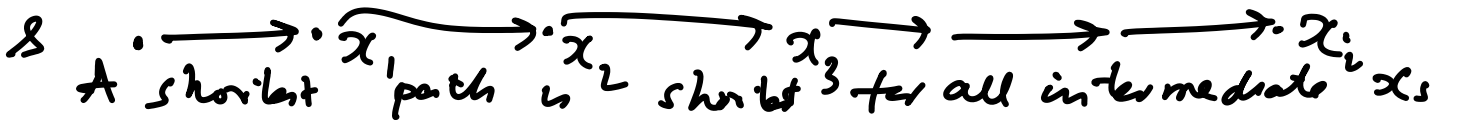
the operation 'shortcut'



$\delta(y)$: shortest path distance from s to y

$$\delta(y) \leq \delta(x) + w$$

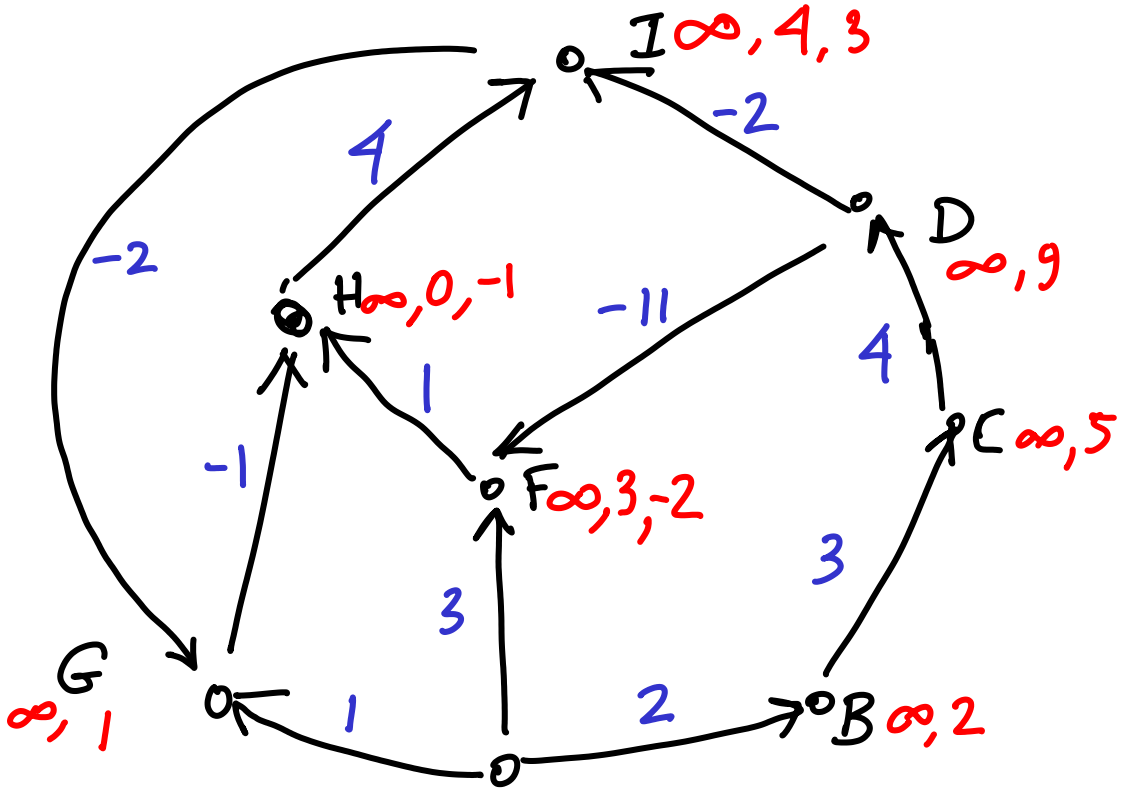
$D(x)$: upper bound on $\delta(x)$, initially ∞



Round 2, 3, 4, 5, 6, 7

$|V| = n$

$|E| = m$



Bellman-Ford

Observation: If the shortest path to vertex V has i edges, then it takes i rounds for $D(V) = \delta(V)$.

So BF algorithm runs in $O(m \cdot n)$ steps

B.F algorithm can also be used to detect negative cycles.

Dijkstra's algorithm works only on non-negative weights. It discovers shortest paths in increasing order of shortest path distances starting from source (= 0).

The correctness can be proved by a simple induction on the above property. In any iteration, the vertex with the smallest label has the correct shortest path distance

Proof (by contradiction): If not, then clearly the predecessor doesn't have the right distance, and so the induction hypothesis fails

Note that because of +ve weights, the $\delta(u) < \delta(v)$ for any edge $u \to v$