Administrivia:

HW0 due now (5 pm)

HW 1 due next Tue
→ 8pm
- Groups ≤ 3
- Submission
  1 from each group
  + form listing other group members
  per problem

Dynamic Programming

Memo(X):
if Ans(X) unknown
  Preprocess(X)
  for every subproblem Y
    Memo(Y)
  Ans[X] ← Postprocess(X)
return Ans[X]

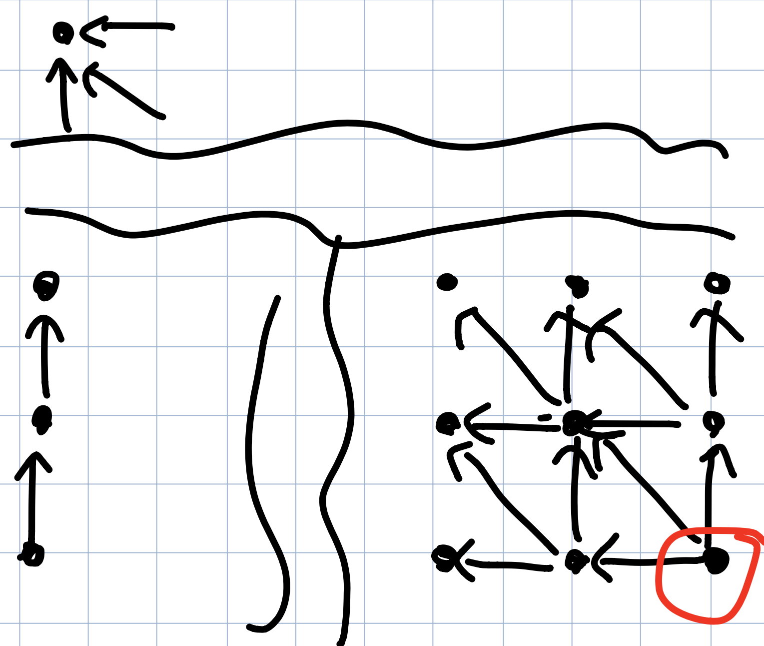
↑ DFS of the recursion dag

DFS(v): ← DAG
if v unmarked
  Previsit(v)
  for each $v \to w$
    DFS(w)
  Postvisit(v)
  mark v

$$
\begin{tikzcd}
F_n \arrow[r] \arrow[rr, bend left] & F_{n-1} \arrow[r] \arrow[rr, bend right] & F_{n-2} \arrow[r] \arrow[rr, bend left] & F_{n-3} \arrow[r] & F_{n-4} & \cdots \arrow[rr, bend right] & F_2 \arrow[r] \arrow[r, bend left, shift left=2] & F_1 & F_0
\end{tikzcd}
$$

Information flows "backward" thru the graph.

Edit



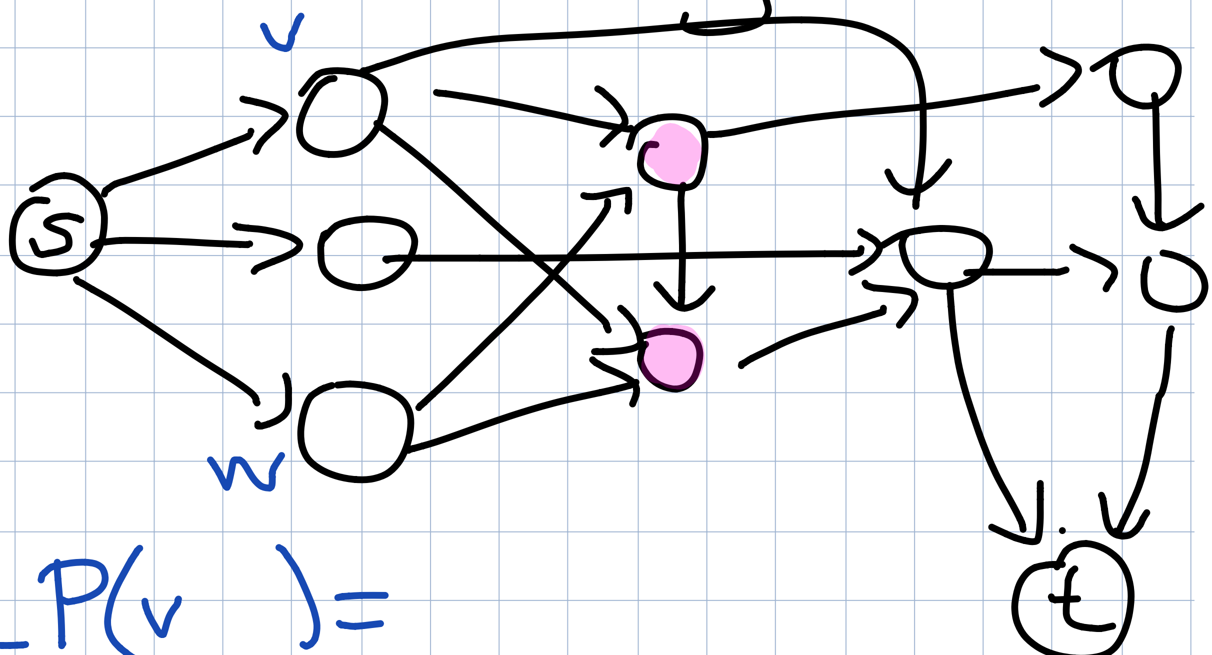
Prep: num ← 0

DFS(v):

if v unmarked
    Previsit(v)
    for each v → w
        DFS(w)
    label(v) ← num++
    mark v

reverse top sort

Longest Path from s to t in a dag



$LP(v) =$ #edges in longest path from v to t

$$LP(v) = \begin{cases} 0 & \text{if } v = t \\ \max\limits_{v \to x} \left(1 + LP(x)\right) & \end{cases}$$

$\max \emptyset = -\infty$

```
LongestPath(v, t):
  if v = t
      return 0
  if v.LLP is undefined
      v.LLP ← ~~∞~~ −∞
      for each edge v→w
          v.LLP ← max{v.LLP, ℓ(v→w) + LongestPath(w, t)}
      return v.LLP
```

$O(V+E)$

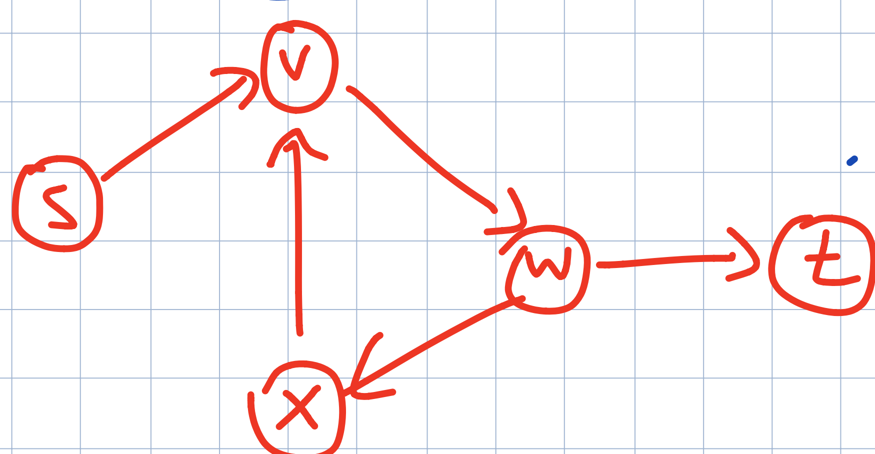
```
LongestPath(s, t):
  for each node v in reverse topological order
      if v = t
          v.LLP ← ~~∞~~ 0
      else
          v.LLP ← ~~∞~~ −∞
          for each edge v→w
              v.LLP ← max{v.LLP, ℓ(v→w) + w.LLP}
  return s.LLP
```

Shortest Path from s to t

Dijkstra $\longrightarrow O(E + V\log V)$

dist(v, w, i) = shortest path distance $v \rightsquigarrow w$ using $\leq i$ edges

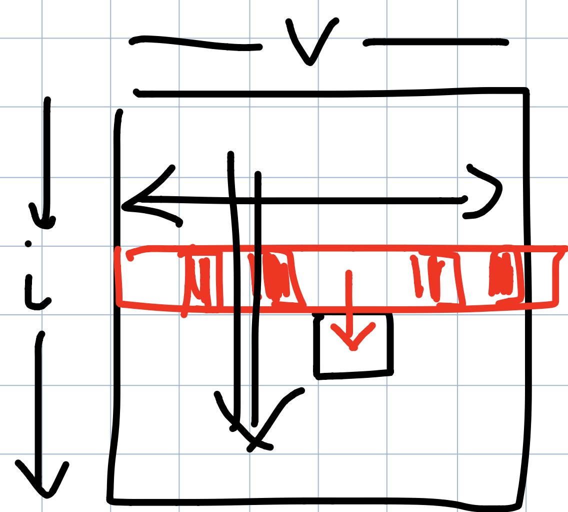
$$\text{dist}(v,w,i) = \begin{cases} 0 & \text{if } v = w \\ \infty & \text{if } v \neq w \text{ and } i = 0 \\ \min\limits_{v \to x} \left\{ \ell(v \to x) + \text{dist}(x, w, i-1) \right\} \end{cases}$$



$dist_i(v)$ = sh. path distance from s to v

$$dist_i(v) = \begin{cases} 0 & \text{if } i = 0 \text{ and } v = s \\ \infty & \text{if } i = 0 \text{ and } v \neq s \\ \min \left\{ \begin{array}{c} dist_{i-1}(v), \\ \min\limits_{u \to v \in E} (dist_{i-1}(u) + w(u \to v)) \end{array} \right\} & \text{otherwise} \end{cases}$$

Memoize:



```
SHIMBELDP(s)
  dist[0, s] ← 0
  for every vertex v ≠ s
      dist[0, v] ← ∞
  for i ← 1 to V − 1
      for every vertex v
          dist[i, v] ← dist[i − 1, v]
          for every edge u→v
              if dist[i, v] > dist[i − 1, u] + w(u→v)
                  dist[i, v] ← dist[i − 1, u] + w(u→v)
```

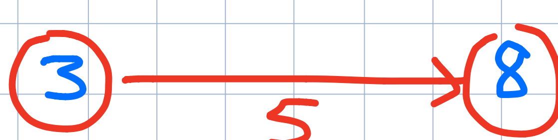
$O(V E)$

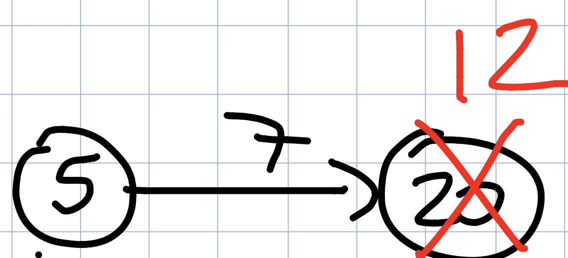
$\underline{\text{SHIMBELDP2}(s)}$

$\quad dist[0,s] \leftarrow 0$
$\quad$ for every vertex $v \neq s$
$\qquad dist[0,v] \leftarrow \infty$
$\quad$ for $i \leftarrow 1$ to $V-1$
$\qquad$ for every vertex v
$\qquad\quad dist[i,v] \leftarrow dist[i-1,v]$
$\qquad$ for every edge $u \rightarrow v$
$\qquad\quad$ if $dist[i,v] > dist[\mathbf{i},u] + w(u \rightarrow v)$
$\qquad\qquad dist[i,v] \leftarrow dist[\mathbf{i},u] + w(u \rightarrow v)$

$\underline{\text{SHIMBELDP3}(s)}$

$\quad dist[s] \leftarrow 0$
$\quad$ for every vertex $v \neq s$
$\qquad dist[v] \leftarrow \infty$
$\quad$ for $i \leftarrow 1$ to $V-1$
$\qquad$ for every edge $u \rightarrow v$
$\qquad\quad$ if $dist[v] > dist[u] + w(u \rightarrow v)$
$\qquad\qquad dist[v] \leftarrow dist[u] + w(u \rightarrow v)$





RELAX($u \to v$):

$dist(v) \leftarrow dist(u) + w(u \to v)$

$pred(v) \leftarrow u$

$$\max S = \begin{cases} -\infty & S = \emptyset \\ \max\{x, \max S \setminus x\} & \end{cases}$$

$$S = \{1\}$$

V times:
Relax ALL the edges!

Hyperbole and a Half