Admin:

HW1 revision #2 → (a)
→ (b) ← correctness: greedy notes

$\boxed{\text{DP rubric}}$

$\boxed{\text{3: English}}$ incl. top level call

pseudocode {
4: recurrence
3: iterative

Dynamic Prog. on Trees ≈ Postorder traversal
Depth-First search

```
Memoize(x):
  if value[x] is undefined
      initialize value[x]
      for all subproblem y of x
          Memoize(y)
          update value[x] based on value[y]
      finalize value[x]
  return value[x]
```

Graph is defined implictly by a recursive function:

- Vertices = subproblems
- Edges = dependencies

```
DepthFirstSearch(v):
  if v is unmarked
      mark v
      PreVisit(v)
      for each edge v→w
          DFS(w)
          ((work))
      PostVisit(v) ←
```

GIVEN an explicit directed graph

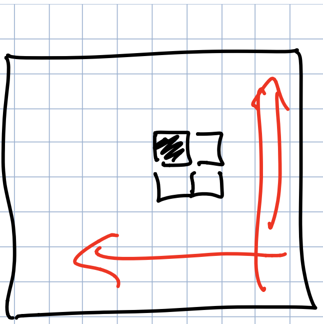
Proper recurrence ⟷ dag ⇐ correctness

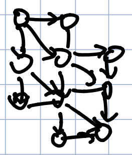
Memoization ⟷ marking ⇐ efficiency

Evaluation order ⟷ (rev) topological order

= postorder

SCS

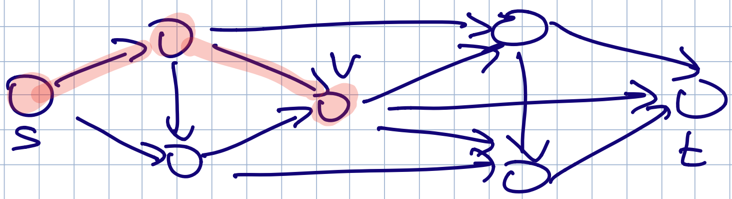




Longest Path(G,s,t)   G is a dag ← else NP-hard!

↳ length of the longest path from s to t in G

Seq of decisions: Where next?



Real recursive prob:

LP(v) = length of longest path from v to t in G

Recurrence:

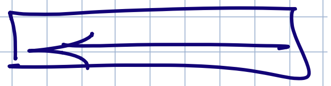
$$LP(v) = \begin{cases} 0 & \text{if } v = t \\ \max\{LP(w)+1 \mid v \to w \in E\} \end{cases}$$

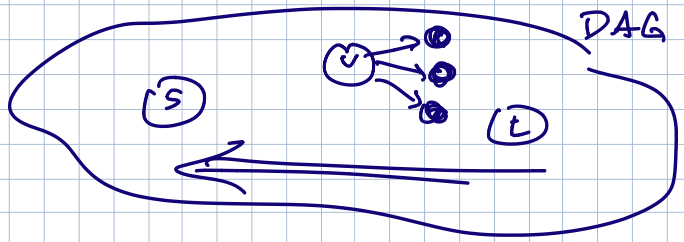
$$\max \emptyset = \cancel{\emptyset}\ -\infty$$

Memoize:
add a field v.LP to every vertex record.

If G is adj list/adj matrix ⇒ ARRAY!

Evaluation order: $\boxed{\text{rev top order}}$ ← DFS + reindex





Equivalently: POSTORDER by DFS

Time = $\boxed{O(V+E)}$

① Recurrence + dyn prog

② DAG + longest path