To DSC decomposition
shortest paths
  BFS, DAG, Dijkstra
Bellman-Ford

\[ \text{SCC}(v) = \text{reach}(v) \cap \text{reach}^{-1}(v) \]

Given \( v \), can find \( \text{SCC}(v) \) in \( O(V+E) \) time
for all \( v \in V \)
\[ \text{DFS}(v) \]
\[ \text{DFS}^{-1}(v) \]
Compute intersection

0. Every SCC has exactly one node \( w \) with parent not in same SCC in a DFS tree
1. DFS from any node in sink component returns all nodes in that SCC and no others
2. Last vertex in post-order of \( G \) is in source component
3. Last vertex in post-order of \( \text{rev}(G) \) is in the sink component

- do post order of \( \text{rev}(G) \)
- DFS on node \( w \) with highest post-order
- repeat for any unvisited nodes, taking highest node as DFS
**KosarajuSharir(G):**

\[ S \leftarrow \text{new empty stack} \]

for all vertices \( v \)

unmark \( v \)

\[ v.root \leftarrow \text{NONE} \]

\{(Phase 1: Push in postorder in rev(G))\}

for all vertices \( v \)

if \( v \) is unmarked

\[ \text{PUSHPostRevDFS}(v, S) \]

\{(Phase 2: DFS again in stack order)\}

while \( S \) is non-empty

\[ v \leftarrow \text{POP}(S) \]

if \( v.root \) = \text{NONE}

\[ \text{LABELONEDFS}(v, r) \]

**PUSHPostRevDFS(v, S):**

mark \( v \)

for each edge \( u \rightarrow v \) \text{ (Reversed!)}

if \( u \) is unmarked

\[ \text{PUSHPostRevDFS}(u, S) \]

**PUSH(v, S)**

**LABELONEDFS(v, r):**

\[ v.root \leftarrow r \]

for each edge \( v \rightarrow w \)

if \( w.root \) = \text{NONE}

\[ \text{LABELONEDFS}(w, r) \]

---

**Figure 6.16.** The Kosaraju-Sharir strong components algorithm
Shortest paths
- Given source find shortest path from $s$ to $t$
- Given source and targets find shortest path to all targets $t$
- Given source $s$, find shortest path to all targets $t$ (SSSP)

Shortest path tree rooted at source $s$

$$\text{SP}(s, u) \leq \text{SP}(s, v) + \ell(v, u)$$

$$\text{DIST}[u] \rightarrow \text{tentative SP length from } s \text{ to } u$$

$$\text{DIST}[s] = 0$$

$$\text{DIST}[u] = \infty \text{ for all } u \neq s$$

**Relax** edge $(v, u)$

$$\text{DIST}[u] \geq \text{DIST}[v] + \ell(v, u)$$

**BFS** if edges are unweighted
**BFS(s):**
- InitSSSP(s)
- Push(s)

  while the queue is not empty
  
  \[ u \leftarrow \text{Pull()} \]
  for all edges \( u \rightarrow v \)
  
  \[ \begin{align*}
    & \text{if } \text{dist}(v) > \text{dist}(u) + 1 \\
    & \quad \langle\text{if } u \rightarrow v \text{ is tense}\rangle \\
    & \quad \text{dist}(v) \leftarrow \text{dist}(u) + 1 \\
    & \quad \langle\text{relax } u \rightarrow v\rangle \\
    & \quad \text{pred}(v) \leftarrow u \\
    & \quad \text{Push}(v)
  \end{align*}\]

**InitSSSP(s):**
- dist(s) \( \leftarrow 0 \)
- pred(s) \( \leftarrow \text{NULL} \)

  for all vertices \( v \neq s \)
  
  \[ \begin{align*}
    & \text{dist}(v) \leftarrow \infty \\
    & \text{pred}(v) \leftarrow \text{NULL}
  \end{align*}\]

**SSSP on DAGs**

\[ \text{SP}(s, u) \leq \text{SP}(s, v) + \ell(v, u) \]

\[ \text{SP}(s, u) = \min_{v \rightarrow u} \text{SP}(s, v) + \ell(v, u) \]

\[ \text{SP}(s, s) = 0 \]

\[ \text{SP}(s, s) = \infty \text{ for all } t \neq s \]
For \( u \in G \), topo order of \( G \):

\[
SP(s, u) = \min_{v \in S} (SP(s, v) + E(u, v))
\]

\( O(u + E) \)

\( s \rightarrow a \)

\( Dijkstra \)

Finished nodes -> have exact SP length

Unfinished nodes -> don’t """

Relax edges from finished to unfinished

Closest unfinished node can be marked finished

\( Dijkstra \) mark all nodes \( \{ s \} \) unfinished

while there are unfinished nodes:

- take smallest unfinished node \( v \) \( \{ Dist \} \)

  - mark as finished

  - relax all edges from \( v \) Ex relax
\( O(n^2) \quad O(n^2 + E) \quad \text{Eis } O(n^2) \)