**Shortcut Paths**

**Input:** Directed graph $G = (V, E)$

- wt on edges (positive)
- $\emptyset \in V$

**Output:** Shortest path tree rooted at $\emptyset$.

**Dijkstra (G, $\emptyset$)**

for each $u \in V$ do

\[ d[u] \leftarrow \infty \]
\[ \pi[u] \leftarrow \text{NIL} \]

\[ d[\emptyset] \leftarrow 0 \]

$S \leftarrow \emptyset$

**Build Heap using $d[.]$ values.**

while $S \neq V$ do

\[ \Theta(n^2) \]

$\emptyset \leftarrow u \leftarrow \text{vertex in } V \setminus S \text{ with the smallest } d[.]$

\[ m + n + n - 2 + \ldots + 1 = \Theta(n^2) \]

ExtractMin
\( O(n^2) \) – \( S \leftarrow S \cup \{ u \} \)

\[ \text{for each } v \in \text{N}(u) \text{ s.t. } v \in V \text{ s.t. } \]

\[ \begin{cases} \text{if } d[v] > d[u] + w_{uv} \text{ then} \\ d[v] \leftarrow d[u] + w_{uv} \\ \pi[v] \leftarrow u \end{cases} \]

Running time: \( O(n^2) \)

New running time (using min-heap):

\[ O(n \log n + m) \]

\[ O(n \log n + m \log n) \]

\[ m \geq n \Rightarrow O(m \log n). \]

Correctness: Induction on \(|S|\).

IH: let \( k \geq 1 \) be an integer. Assume that Dijkstra computes shortest path.
correctly for all vertices in $S$, when $|S| = k$.

**BC**: $|S| = 1$, $S$ contains $v$, $d(v) = 0$.

**IS**: Want to prove the claim either $|S| = k + 1$.

Let $v$ be the $(k+1)^{th}$ vertex brought into $S$ via the edge $(u,v)$. Then $d(v)$ is the shortest path length as computed by Dijkstra, i.e.,

$d(v) = d(u) + uv$. Assume for contradiction that this is not the right answer. Let $P$ be the actual
This is a contradiction because if $\{y\} \subseteq T_2$ then $y$ cannot be an element of $\mathcal{F}$. Furthermore, $\mu(y) > \mu_P(x)$ for all $x \in T_1$. Therefore, $\mu(y) > \mu_P(y)$, which is a contradiction. The shortest path from $x$ to $y$ is...
the (KPI)™ metric to be brought into.

Fix: Take the "most -ve edge" &
add a value to make everything true.
Then run Dijkstra. X

Larry: Fix is bogus!

**Strongly Connected Components (SCC)**

**Input:** Directed graph \( G = (V, E) \).

**Output:** All SCCs of \( G \).

\( H \) is a SCC of \( G \) i\( \iff \)

- \( H \) is a subgraph of \( G \)
- \( u, v \) in \( H \), \( u \rightarrow v \) and \( v \rightarrow u \).
Observations:
1. The SCCs remain the same when we reverse the direction of all edges.
2. $G_{SCC} = (V_{SCC}, E_{SCC})$

Each vertex in $V_{SCC}$ corresponds to a SCC in $G$ and an edge from $C_0$
C' in \( G^{\text{sc}} \) exists if there is an edge from a little vertex in C to a little vertex in C'.

Property of \( G^{\text{sc}} \): DAG.

\( G^{\text{sc}} \) must have a sink vertex.

**Claim:** If we do DFS \((G)\) then the vertex with the smallest \( f(i) \) belongs to the sink node in \( G^{\text{sc}} \).

**Alg:**
1. DFS \((G)\)
2. For each edge in \( G \)
Reversing the answer:

2. Do DFS (G), but in the main loop of DFS (i.e., while choosing a new root vertex in the DFS forest), process the vertices in order of their finishes.

3. Vertices in each tree will make SCCs.

Claim: If we do DFS (G), then the vertex that finishes last will always belong to the source vertex in G. 

AE: (Kosaraju)

1. DFS (G) → O(n+m)

for each u ∈ V do
  color[u] ← white
2. for all vertices u
  if color[u] == white
    DFS(u)
2. Compute $G^T \rightarrow o(n+m)$

3. $DFS(G^T)$, but in the main loop of $DFS$
   process vertices in $\Delta$ order $\Delta H$.

4. Vertices in each tree of $DFS$ make up
   $S(t)$.
1. $G^T$
2. DFS ($G^T$)
3. DFS ($G$)