Shortest Paths

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

wt on edges (positive)

$s \in V$

Output: Shortest path tree rooted at $s$.

Dijkstra $(G, s)$

For each $u \in V$ do

D$(u)$

$0(1)$

$d[u] \leftarrow \infty$

$\pi[u] \leftarrow NIL$

$d[s] \leftarrow 0$

$S \leftarrow \emptyset$

while $S \neq V$

BuildHeap$(S)$

$O(n)$

$O(n^2) \rightarrow u \leftarrow \text{vertex in } V \setminus S \text{ with the smallest } d[u]$
\[ O(n) \rightarrow S \leftarrow S \cup \{v\} \]

for each \( u \in N(v) \) s.t. \( v \in V \) \( S \) do

\[ \Sigma_{u \in N(v)} d(u) = O(m) \]

if \( d(v) > d(u) + w_{uv} \) then

\[ \begin{align*}
    d(v) &\leftarrow d(u) + w_{uv} \\
    \pi(v) &\leftarrow u
\end{align*} \]

Decrease key \( \Sigma_{u \in N(v)} d(u) \cdot y(u) = O(m \gamma n) \)

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

Build a heap with the \( d[\cdot] \) values being the keys & then use ExtractMin.

New running time: \( O(n \gamma n + m \gamma n) \)

Support \( m \gg n \): \( O(m \gamma n) \).

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

I H: Let \( k \geq 1 \) be an integer. Assume that
When $|S| = k$, Dijkstra compute shortest paths from $\emptyset$ to every vertex in $S$ correctly.

**BC**: $|S| = 1$. $S$ contains $\emptyset$ & $d[\emptyset] = 0$.

**IS**: Want to prove the claim when $|S| = k+1$. Let $v$ be the $(k+1)^{th}$ vertex that is brought into $S$. Who buys $wv$? $w$ is the parent of $v$.

![Diagram of a graph with vertices and edges, including labels and annotations.]

Shortest path from $\emptyset$ to $v$ returned by
Dijkstra = \[d[u] = d[u] + \omega_{uv}.

Assume for contradiction that Dijkstra gives a wrong answer for vertex \(v\).

Let \(P\) be a shortest path from \(s\) to \(v\). Let \(x\) be the last vertex on \(P\) before it leaves \(S\). Let \((x,y)\) belong to \(P\). We have:

\[d[u] \geq d[x] + \omega_{xy} + \omega_{Pyv}.

But then \(d[y] < d[u]\), contradicting that \(v\) is the \((\text{1st})\)th vertex to be brought into \(S\). If \(y = v\) then it contradicts line \(v\) is
brought into $S$, i.e., it should not be a bringing $v$, but a bringing $v$.

Fix for -ve edge costs:

Add a large value to every edge $uv$.
Then run Dijkstra.

Jay's claim: the above claim is bogus.

**Strongly Connected Components (SCC)**

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

**Output:** all SCCs of $G$.

$H$ is a SCC of $G$ if:
- $H$ is a subgraph of $G$. 

- \( \forall u, v \in H, u \sim v \) and \( v \sim u \).
- \( H \) is maximal.

**Observation**:

1. By reversing the direction of each edge \( e \) in \( G \), the SCCs stay the same.

2. \( G^{\text{SCC}} = (V^{\text{SCC}}, E^{\text{SCC}}) \)

Each vertex in \( V^{\text{SCC}} \) is a SCC in \( G \).

\( e = (C, C') \in E^{\text{SCC}} \) if there is a
vertices in $C$ that has an edge going to $C'$. 

$G^{\text{SCC}}$ is a DAG.

Thus $G^{\text{SCC}}$ has a sink node.

**Question:** How do we find a little vertex in the sink node?

**Claim:** If we do DFS ($G$) then the vertex with the smallest $f[.]$ must belong to the sink node in $G^{\text{SCC}}$.

**Algorithm:**

1. DFS ($G$) 
2. Return direction of all edges
2. Do DFS (4) again, but in the main loop of DFS, process vertices in \( \sqrt{\bar{f}} \) order if \( f[.] \). In other words, when deciding which vertex to chosen as the root of the next tree, pick one with the smallest \( f[.] \).

Counterexample to the claim:

\[
\begin{array}{cccc}
1/6 & 1/6 & a & b \\
2/3 & 4/5 & 2/3 & 4/5
\end{array}
\]
Can we get a handle to a vertex belonging to the source node of $G_{SCC}$?

Claim: The vertex with the latest finish time must belong to the source node of $G_{SCC}$.

Also (kosaraju)

1. $DFSC(G)$ \(\rightarrow O(n+m)\)

2. Compute $G^T$ \(\rightarrow O(n+m)\)

3. $DFS(G^T)$, but in the main loop, process vertices in the reverse order of $f[.].$

4. Vertices of each tree make one SCC.

12/20. 18/19
You may assume that $G^{sec}$ can be computed in $O(n+m)$ time.