Shortest paths

directed

Given a graph $G = (V, E)$ with weights $w : E \rightarrow \mathbb{R}$

(i) For $s, t \in V$ find the shortest path from $s$ to $t$

(ii) Given $s \in V$ find the shortest path from $s$ to all $v \in V$

Single Source Shortest Path: Dijkstra, Bellman-Ford

(iii) All pair shortest paths APSP

for $s$ to $t$

A path $P$ is a sequence of edges

$e_1 = (u, v_1) \ e_2 = (v_2, v_2) \ e_i = (u_i, v_i)$

such that $v_i = u_{i+1}$ and $u_1 = s$

$v_k = t$

The weight $w(P) = \sum w(e_i)$

Length $(P) = \#$ edges/ships
Nature of the weight function

\[ w: \text{Dijkstra} \quad \Rightarrow \text{non-negative} \quad \text{"Easier to solve"} \]

\[ : \text{negative} \quad \Rightarrow \text{no negative cycles} \]

\[ \Rightarrow \text{negative cycles} \]

Only simple paths are allowed (no vertices repeated)

Lengths \( \leq |V|-1 \)

All pair shortest path (APSP)

Some properties of shortest paths

Subpath optimality

If \( V_0, V_1, V_2, \ldots, V_k \) is a shortest path between \( V_0 = s \) and \( V_k = t \)

Then \( V_i, V_{in}, \ldots, V_k \) is also a shortest path between \( V_i \) and \( V_k \)
The shortest path distance \( \delta(v) \): Shortest path distance from \( s \) to \( v \)

\[
\delta(v) = \min \left\{ \delta(v_{ij}) + w_{ij} \right\}
\]

Set of all neighbors \( \mathcal{N} \)

\( v \) s.t. \((v_{i}, v) \in E\)

The actual shortest path (not distance) actually form a tree rooted at \( s \).

Induction for Floyd-Warshall:

\( \rho_{i} \) is a path from \( s \) to \( v \) that does not use any vertex numbered greater than \( i \), not including \( s \), \( v \).
\[ V = \{1, 2, \ldots, n\} \]

\[ P_1 = \begin{bmatrix} 1 & 3 & 4 \end{bmatrix} \quad P_2 = \begin{bmatrix} 0 & 2 & 3 \end{bmatrix} \]

\[ P_1 \notin P^4 \quad P_2 \notin P^3 \]

\[ P_1 \in P^3 \quad P_2 \in P^2 \]

Floyd-Warshall approach is to begin with \( P^0 \): paths consist of single edges. Subsequently, we construct \( P^1, P^2, \ldots, P^n \).

Suppose we have computed \( P^k \).

\( P^{(k)}(i, j) \) denotes the shortest path between vertices \( i, j \) that doesn't use vertex higher than \( k \).

\[ P^{(0)}(i, j) = w(i, j) \quad \text{if there is an edge } (i, j) \]

\[ \infty \quad \text{if there is no edge} \]
We have \( p^{(k)}(i,j) \neq p^{(k+1)}(i,j) \) for \( i,j \in V \), as the shortest path does not go through \( k+1 \).

\[
\begin{align*}
P^{(k+1)}(i,j) &= P^{(k)}(i,j) \quad \text{if } p^{(k+1)}(i,j) = p^{(k)}(i,j) \\
&= \min \left\{ p^{(k)}(i,j), \left[ p^{(k)}(i,k+1) + p^{(k)}(k+1,j) \right] \right\}
\end{align*}
\]

D.P. for Floyd Warshall computes \( n^3 \) entries in increasing order of \( k \) and each entry takes \( O(1) \) time,

\[ \Rightarrow O(n^3) \text{ time} \]

\( O(n^2) \) space since only Table \( k \) is needed.
How to get the actual paths?

How to store all paths?

\[ i \rightarrow \text{some} \rightarrow j : \text{dest} \]

\[ i \rightarrow (i,j) \rightarrow j \]

With all look ups we can reconstruct the shortest