Shortest Distance

$G = (V, E)$ $\quad w : E \to \mathbb{R}$.
$\qquad\qquad\qquad\qquad\qquad\quad \mathbb{Z}_+$

↳ Edge weights.

Qu: Given $G = (V, E)$, and nodes $s, t \in V$, what is the $\underbrace{\text{length of the shortest path}}_{\text{shortest distance.}}$ between s and t?

If G is unweighted (or has uniform weights), BFS gives us shortest distance.

(Greedy algorithms)

Dijkstra's algorithm

Input: $G = (V, E)$, L, s ← start node

L ↑ list of weights on edges.

Output: List of shortest distances from s to every other node in the graph.

S: Set of nodes to which the shortest distance from s is already computed.

initialization

- $S = \{s\}$, $d(s) = 0$
- For every other vertex $v \in V$: $d(v) = \infty$

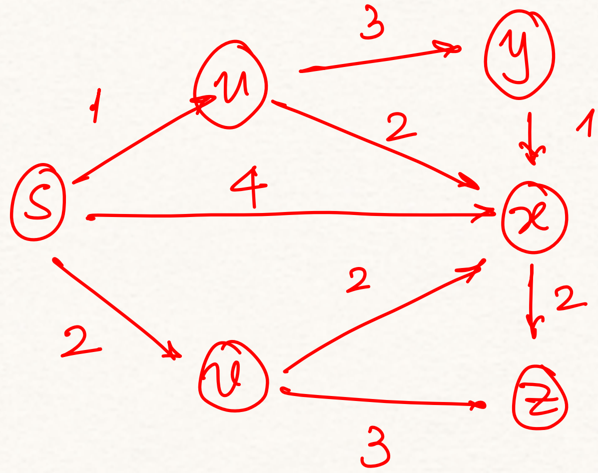
O(n) array updates.

While $S \neq V$:

Select a vertex v with at least one edge from S s.t $d'(v)$ is minimized for v among all nodes in $N(S)$ where

$$d'(v) = \min_{(u,v) \in E, \text{ and } u \in S} \{ d(u) + l_{u,v} \}.$$

Add v to S and set $\underline{\underline{d(v)}} = d'(v)$



$d[v] = \infty \ \forall v \neq s.$

$S' = \{s\}$

$\hookrightarrow N(S') = \{u, x, v\}.$

$d'(u) = d(s,s) + l_{s,u} = 1$

$d'(v) = d(s,s) + l_{s,v} = 2$

$d'(x) = 4.$

$S' = \{s, u\}.$

$N(S') = \{x, v, y\}$

$d'(x) = \min\{ d(s,u) + l_{u,x} , d_{s,s} + l_{s,x} \}$

$= \min\{ 1+2, 0+4 \} = 3$

$d'(v) = 2$

$d'(y) = \{ d(u) + l_{u,y} \} = 4$

$S' = \{s, u, v\}.$

$d(u) = 1, d(v) = 2$

$d(s) = 0$

$N(S') = \{x, y, z\}.$

$d'(y) = d(u) + l_{u,y} = 4$

$d'(z) = 5$

$d'(x) = \min\{ d(v) + l_{v,x},$

$d(u) + l_{u,x}, l_{s,x} \}$

$= 3$

$S' = \{s, u, v, x\}.$

Obs: distances of elements in S' do not get updated later.

Maintain a Priority Queue for each $u \notin S$ where key value $= d'[u]$

One extract Min per iteration.

$O(m)$ changeKey operations overall.

$O(n) + n$ ExtractMin op $+ O(m)$ Change key $+ O(n)$ overhead.

Correctness:

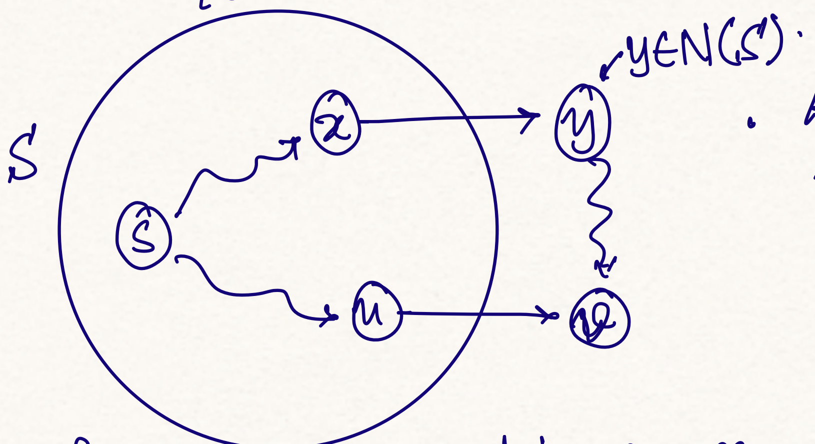
Lemma: Consider the set S at an arbitrary point of the execution of the algorithm. For all $u \in S$, $d(u)$ is the shortest $s \rightsquigarrow u$ distance.

<u>Proof</u>: Induction on S.

Base case: $|S| = 1$. $d(s) = 0$ // Trivial.

I.H: Statement of the lemma is true for $|S| = k$ for some $k \geq 1$.

Algorithm picks $v \in N(S)$ s.t $d'(v)$ is minimum over $\{d'(u) : u \in N(S)\}$ and set $\underline{d(v)} = \underline{d'(v)}$.



- Algorithm picked v over y

$d'(v) \leq d'(y)$.

For the sake of contradiction, let us assume that the shortest $s \rightsquigarrow v$ path is through $\underbrace{s \rightsquigarrow x - y \rightsquigarrow v}$.

$$\underbrace{\underbrace{d(s,x) + l_{x,y}}_{d'(y)} + l_{y,v}}_{\text{dist of path } s \rightsquigarrow x - y \rightsquigarrow v.} = d'(y) + l_{y,v} \geq d'(v) + l_{y,v} \geq d'(v) + 1 = d(v) + 1.$$

This contradicts our assumption that $s \rightsquigarrow x - y \rightsquigarrow v$ is shorter than $s \rightsquigarrow x - v$.