Shortest paths

- $G = (V, E)$
  ↳ Edge weights / lengths / distances.

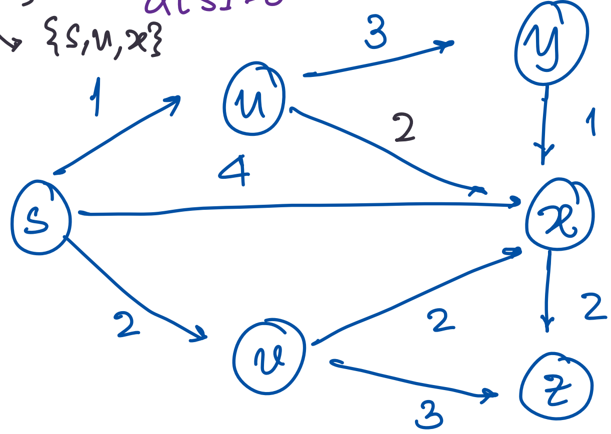
$d'[y]=4$, $d'[v]=2$, $d'[x]=3$

Visited set $= \{s, u\}$ → $\{s, u, x\}$

↓

$\{s, u, v\}$

$d[u]=1$
$d[s]=0$



Start from s and pick nearest neighbour. Mark s as "visited".

Explore all the neigh

$d'[x] = \min\{\underbrace{d[u]}_{=1}+2,\ \underbrace{d[s]}_{=0}+4\}$.

Question: Given $G=(V,E)$, $s,t \in V(G)$, what is the shortest distance between s and t? What is that path.

Dijkstra's algorithm (G, L, s)

start node.

list of weights/lengths of edges $\in \mathbb{R}_{\geq 0}$.

Output: Return a list of shortest distances of every node from s.

$S \leftarrow \{s\}$; $d(s)=0$ // S is the set of all "visited" nodes.

For every $v \neq s \in V$: // d maintains shortest distance from s for every node
    $d(v) = \infty$.

While $S \neq V$:
    Select $v \notin S$ w/ at least one edge from S s.t
    $d'(v) = \min_{(u,v)\in E} \{ d(u) + \ell_{u,v} \}$ is minimized. // Greedy

$d[v] = d'(v)$

Add v to S'.

Correctness:

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

Proof: Induction on $|S'|$.

Base case: $|S'| = 1$. $S' = \{s\}$. $d[s] = 0$.

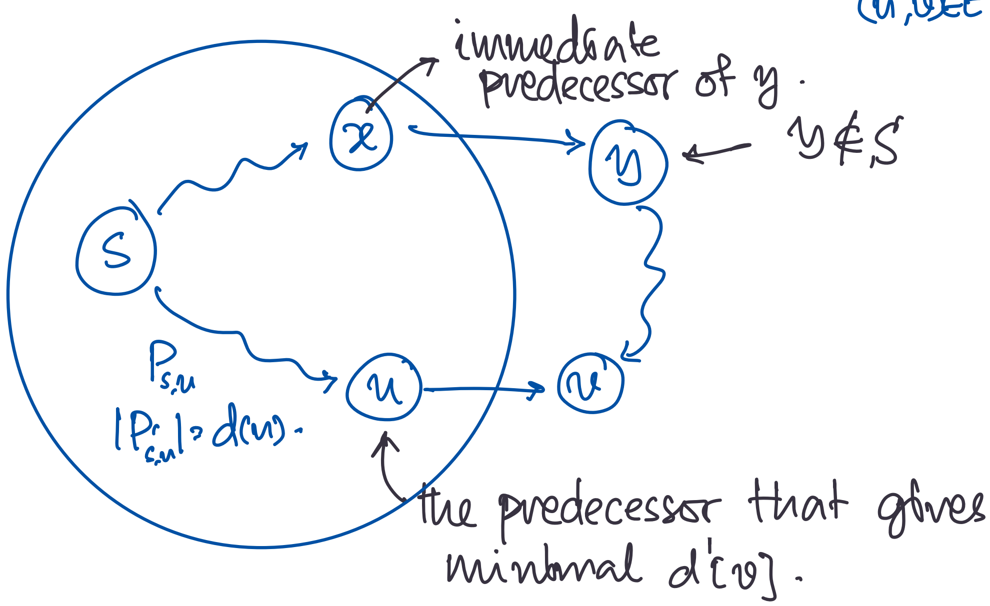
Induction hypothesis: The statement is true for $|S'| = k$ for some $k \geq 1$. $k \leq n-1$.

→ Algorithm picks v s.t $d'[v]$ is the minimum over $\{d'[v'] : v' \in V \setminus S'\}$.

$$d'[v] = \min\{d'[v'] : v' \notin S\}.$$

→ Algo set $d[v]$ to $d'[v]$.

$$d'[v] = \min_{(u,v) \in E}\{d(u) + l_{u,v}\}.$$



Algo computed d' values for all vertices not in S.

↳ Algo picked v over $y \Rightarrow d'[v] \le d'[y]$.

For the sake of contradiction, let the shortest path from s to v be via $s \sim x - y \sim v$. $y \neq v$.

→ Since v was picked no path from s to y was shorter than s to v.

$$\overbrace{s \sim x \to y \sim v}$$

$$\begin{aligned}\text{dist: } & \ell(s,x) + \ell_{x,y} + \ell(y,v) \\ & \geq d[x] + \ell_{x,y} + \ell(y,v) \\ & \geq d'[y] + \ell(y,v) \\ & \geq d'[v] + \ell(y,v)\end{aligned}$$

$\geq$

$$s \sim u \to v$$

$$\begin{aligned} & \ell(s,u) + \ell_{u,v} \\ & = d[u] + \ell_{u,v} \\ & = d'[v]\end{aligned}$$

distance for any other path (say through $y \notin S$) will at least be $d'[y] + \ell(y,v)$.
and if v was picked before y, then $d'[y] \geq d'[v] = d[v]$

$\Rightarrow \cancel{d}[v] = d'[v]$ is in fact the shortest s to v distance.

Implementation and running time:

- $O(n)$ initialization
- Priority queue: maintain $d'[u]$ for every $u \notin S$.
  - ExtractMin operation for every iteration

- ChangeKey $\leftarrow m$ many operations overall.

$O(n)$ + n ExtractMin + m ChangeKey + $O(n)$ overhead.