Shortest Path(s) problem:

I/P : Directed graph $G = (V, E)$

integer $l_e \geq 0 \quad \forall e \in E$

$s \in V$

Output: $\forall t \in V$, shortest s-t path

Given a path P, $l(P) = \sum_{e \in P} l_e$

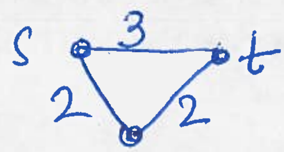
for all s-t path P return $\min_P l(P)$

Special cases: $l_e = c \ \forall e \in E$ for the same c

Can solve by BFS in time $O(m+n)$

Mid term: $O(l_{max}(m+n))$

largest l_e

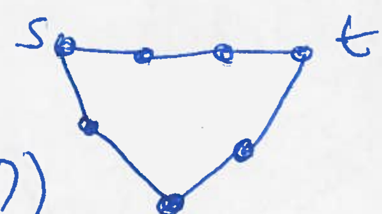


n, m

$\downarrow$

$O(m' + n')$

$= O(l_{max}(m+n))$



$m' \leq l_{max} m$

$n' \leq n + (l_{max} - 1) m$

→ If $l_{max} = O(1)$

→ E.g. $l_{max} = n^{O(1)}$ → # bits to represent l_{max}: $O(\log n)$

Can do basic manipulations on numbers in $O(\log n)$ bits in $O(1)$ time.

Dijkstra's Algo

→ Instead of shortest s-t path, output $d(t)$ = length of shortest s-t path.

Main idea:

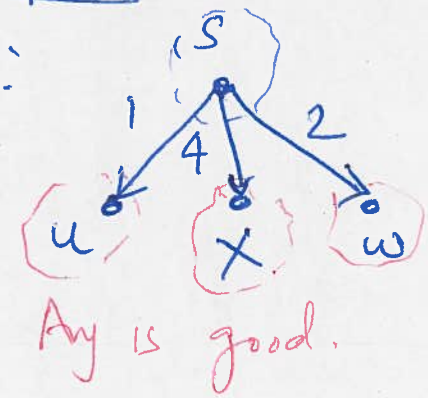
Greedy algo

→ In each iteration, we're going to fix/compute $d(u)$ for one "new" u.

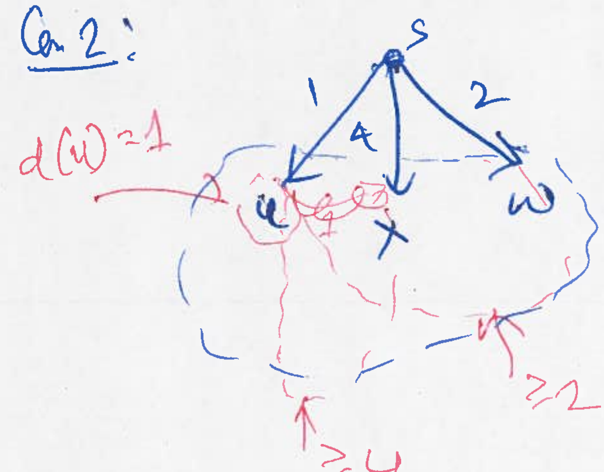
→ **Step 0:** $d(s) = 0$

→ **Step 1:**

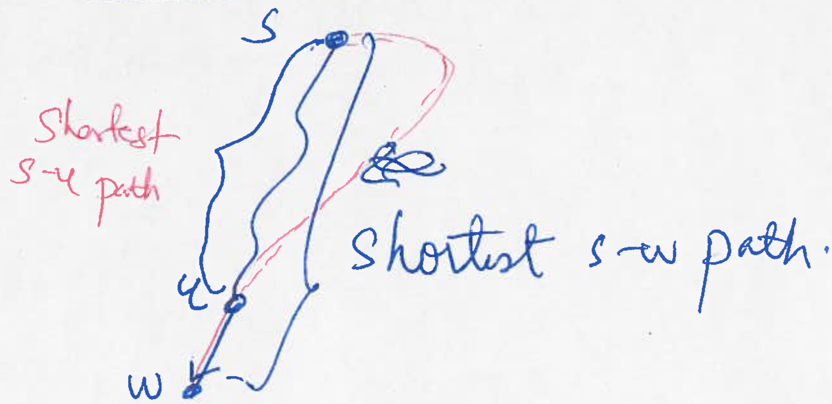
Case 1:

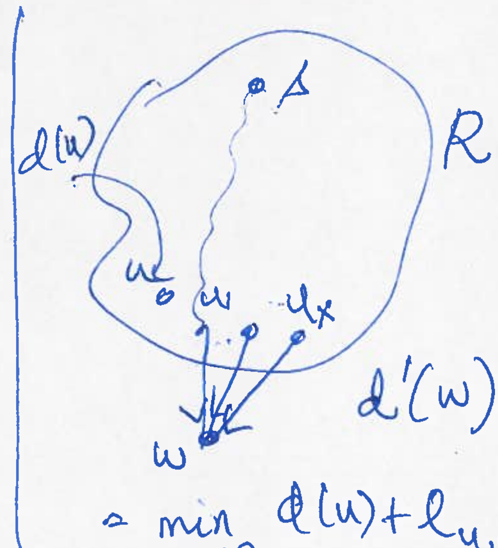


Case 2:

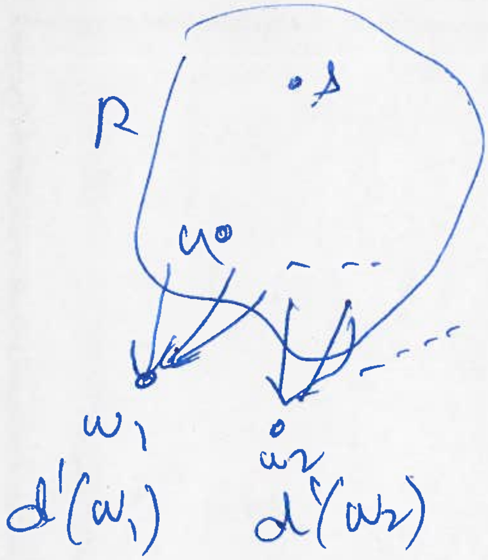


General case:





$= \min\limits_{\substack{u \in R \\ (u,w) \in E}} d(u) + \ell_{u,w}$



Dijkstra's

0. $R \leftarrow \{s\}$, $d(s) \leftarrow 0$

1. While $\exists\, x \notin R$ with $(u,x) \in E$, $u \in R$

   Pick w with the smallest $d'(w)$ value

   Add w to R

   $d(w) \leftarrow d'(w)$