Shortest Path Problem

Oddy

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

$s \in V$

"length" $\rightarrow l_e \geq 0 \quad \forall e \in E$

↑ integer

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

↑ with respect to length of s-t path

$$l(P) = \sum_{e \in P} l_e$$

Simpler version: only output $d(t) \quad \forall t \in V$

↑ length of shortest s-t path.

Special case: $l_e = 1 \quad \forall e \in E \equiv$ same problem HW3 Q3

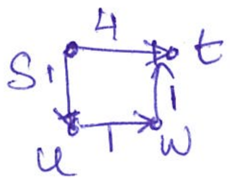
[Run BFS on s & layer # of $t = d(t)$

(also if $l_e = L \ \forall e \quad (L \geq 1)$]

General case: $l_e > 0 \quad \forall e \in E$ | Idea: Reduce this to the case of $l_e = 1 \ \forall e$

Idea: Ignore l_e (i.e. just assume $l_e = 1 \ \forall e$)

& run algo from HW3 Q3.

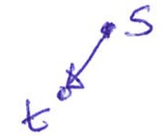


$d(t) = 3$

s, u, w, t

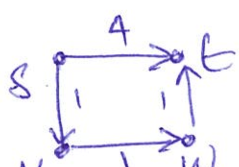
Run

$\xrightarrow{\text{ignore } l_e}$

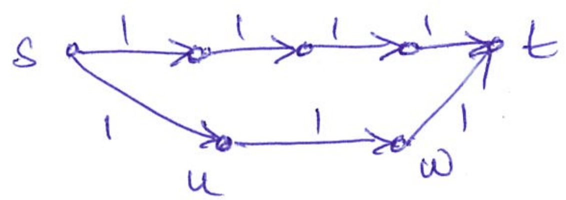


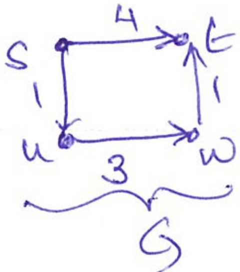
Mistakenly claim $d(t) = 1$.

Idea2: Replace each edge $e \in E$ by a path of length l_e

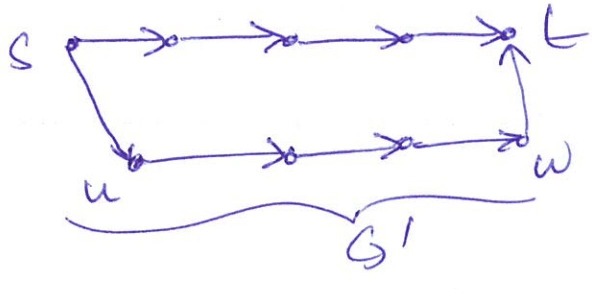


$\longrightarrow$





$\longrightarrow$



Claim: a shortest s-t path in G $\Longleftrightarrow$ equivalent shortest path in G'.

$\Rightarrow$ Run HW 3 Q3 algo on G'.

Correctness: Claim + correctness of HW 3 Q3 algo.

Runtime analysis: $l_{max} = \max_{e \in E} l_e$

Let n' & m' be the # nodes & # edges in G'.

$O(n' + m')$

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

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

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

Recap/Aside: RAM model: unit of space is a register

If you have n items each register has $O(\log n)$ bits.

$\rightarrow$ All basic ops on $O(1)$ registers is $O(1)$ time.

$\rightarrow$ let $l_{max} = n^{100} \Rightarrow 100 \log n$ bits or $O(1)$ many registers

$\longrightarrow$ Input size (# registers) $= O(m+n)$

runtime $\Theta(n^{100}(n+m))$

Assume: $l_{max} : n^{O(1)} \Rightarrow O(1)$ registers for each l_e

$\Rightarrow$ i/p length $= O(n+m)$

WANT: Ideally runtime $O(m+n)$