Algo idea: Replace each edge $e \in E$ by a path of length l_e. (new paths do not share edges/vertices)

$\Rightarrow$ Run HW3 Q3 algo.

claim: a shortest s-t path in G $\Leftrightarrow$ equiv. a shortest s-t path in G'

Correctness: $\rightarrow$ claim + correctness of HW3 Q3 algo

Runtime:

$O(|V'|+|E'|)$ $\quad |V'| = n'$

$O(m'+n')$ $\quad |E'| = m'$

Main Q:

$l_{max} = \max l_e$

$m' \leq l_{max} \cdot m$

$n' \leq l_{max} \cdot n$

G: $O(|V|+|E|)$

$O(m+n)$

$|V| = m$

$|E| = n$

runtime: $O(l_{max} \cdot (m+n))$

if $l_{max} = O(1)$, then fine!

if $l_{max} = n^{100} \rightarrow$ runtime: $O(n^{100}(m+n))$

Aside/Recap: RAM model of computation
unit of space is a register.
if you have n items, each register is $\log(n)$ bits.
adj. list representation: $O(n+m)$ registers.

Q. How many registers do you need to represent $\ell_e \leq n^{100}$

↳ How bits is each register?
↳ $\approx 100 \log n$ bits
$\approx$ 100 registers
$= O(1)$ registers.

Input (# registers): $O(n+m)$
runtime: $O(n^{100}(n+m))$

IDEALLY: want to have $O(n+m)$ runtime for the case $\ell_{max} \leq n^{O(1)}$