Low stretch spanning tree.

Graph distance: G (un)weighted.

distance $d(x,y) = \min_{\text{paths from } x \text{ to } y} \sum_{\text{for all edges in path}} \text{edge weights}$

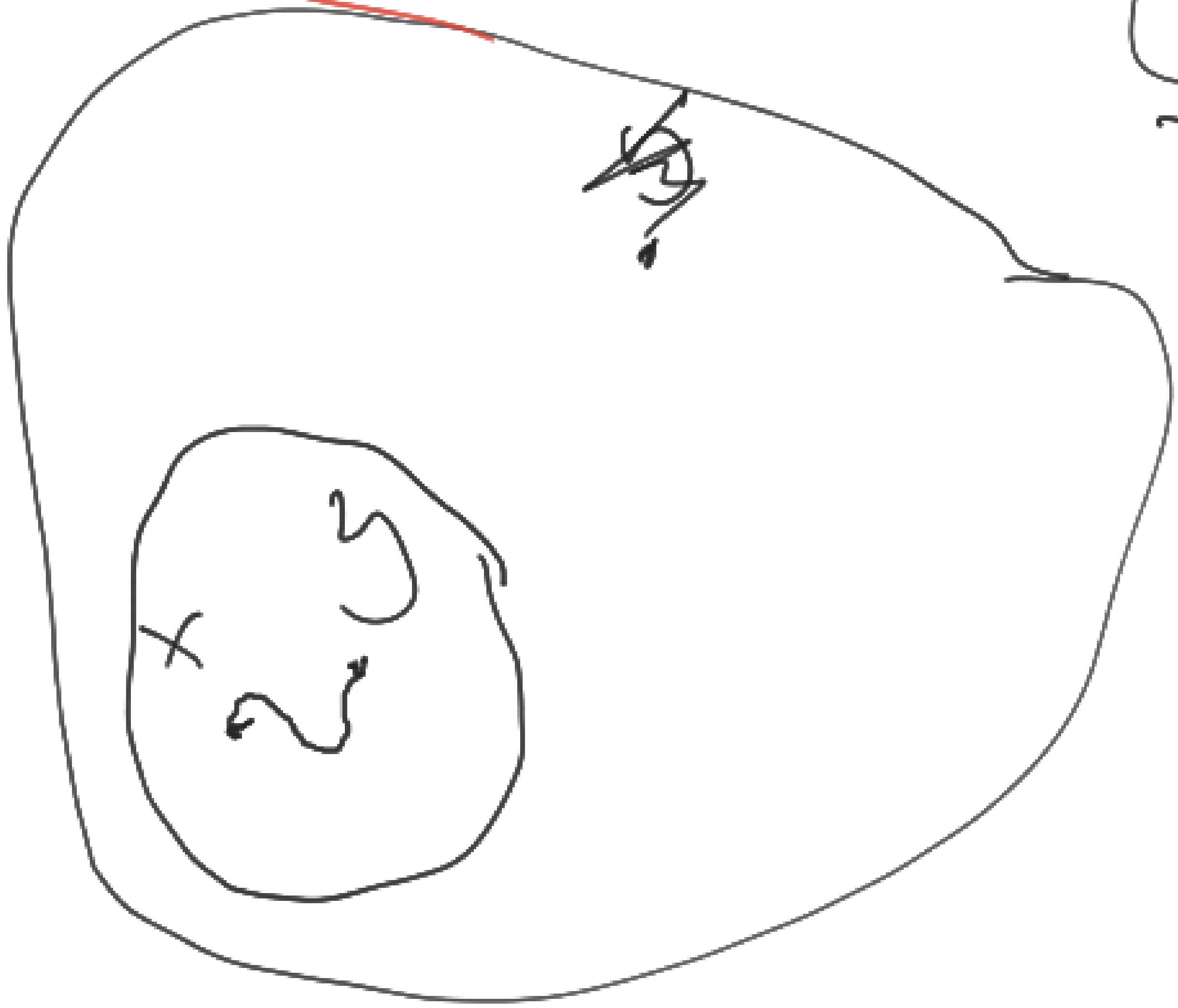
Dijkstra's Algo

compute ~~short~~ shortest path from x to y

(if G has non-negative weights).

Q: Graph is unweighted → True

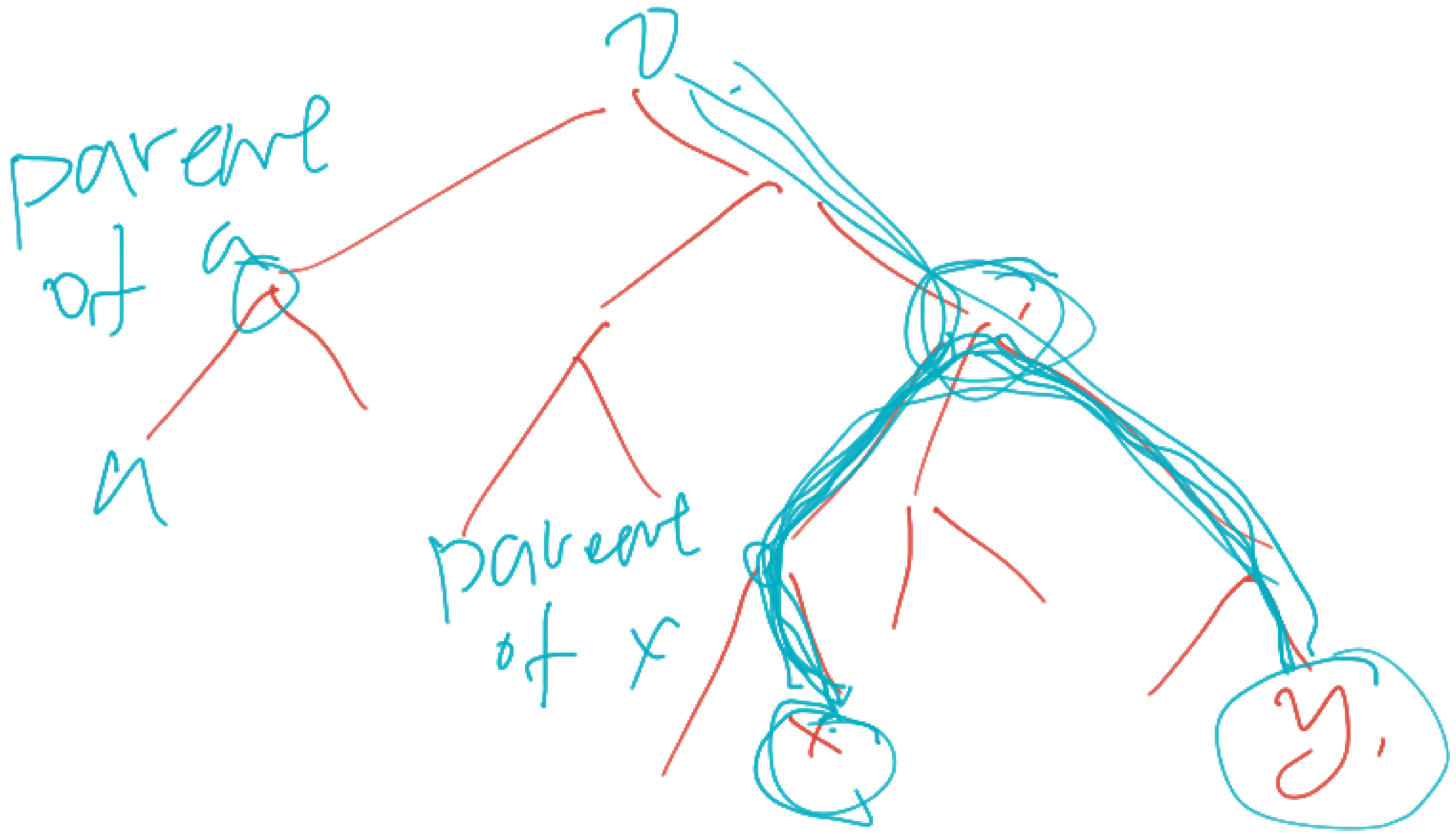
? Can you compute distance $d(x,y)$ in time proportional to $d(x,y)$ $(O(d(x,y)))$.



BFS ✗

$d(x,y) = \log n$.

$O(n)$.



rooted.

each vertex knows its parent in the tree.

Alternatively find path from x to root and path from y to root. $O(d(x,y))$

General Graph

IDEA: Process general graph, produce a tree s.t. distance within the tree is close to the general graph.

routing scheme. → Tree for network

find a good tree

Goal: Find a spanning tree T of G

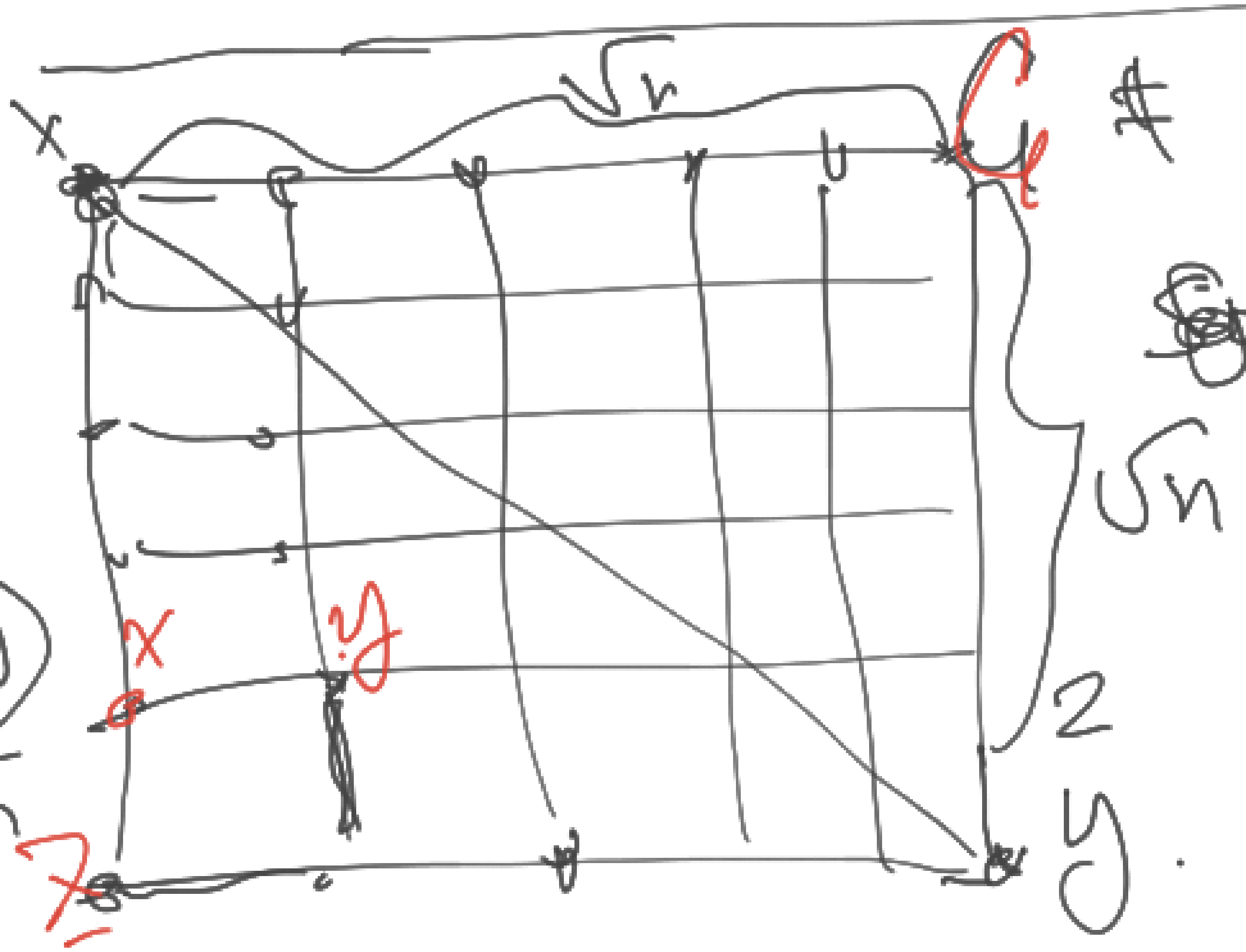
s.t. for "most" vertex pairs (x,y)

$$d_T(x,y) \approx d_G(x,y)$$

$\mathrm{Stretch}(x,y) := 1$

$\mathrm{Stretch}(x,z) = 2\sqrt{n} - 1$

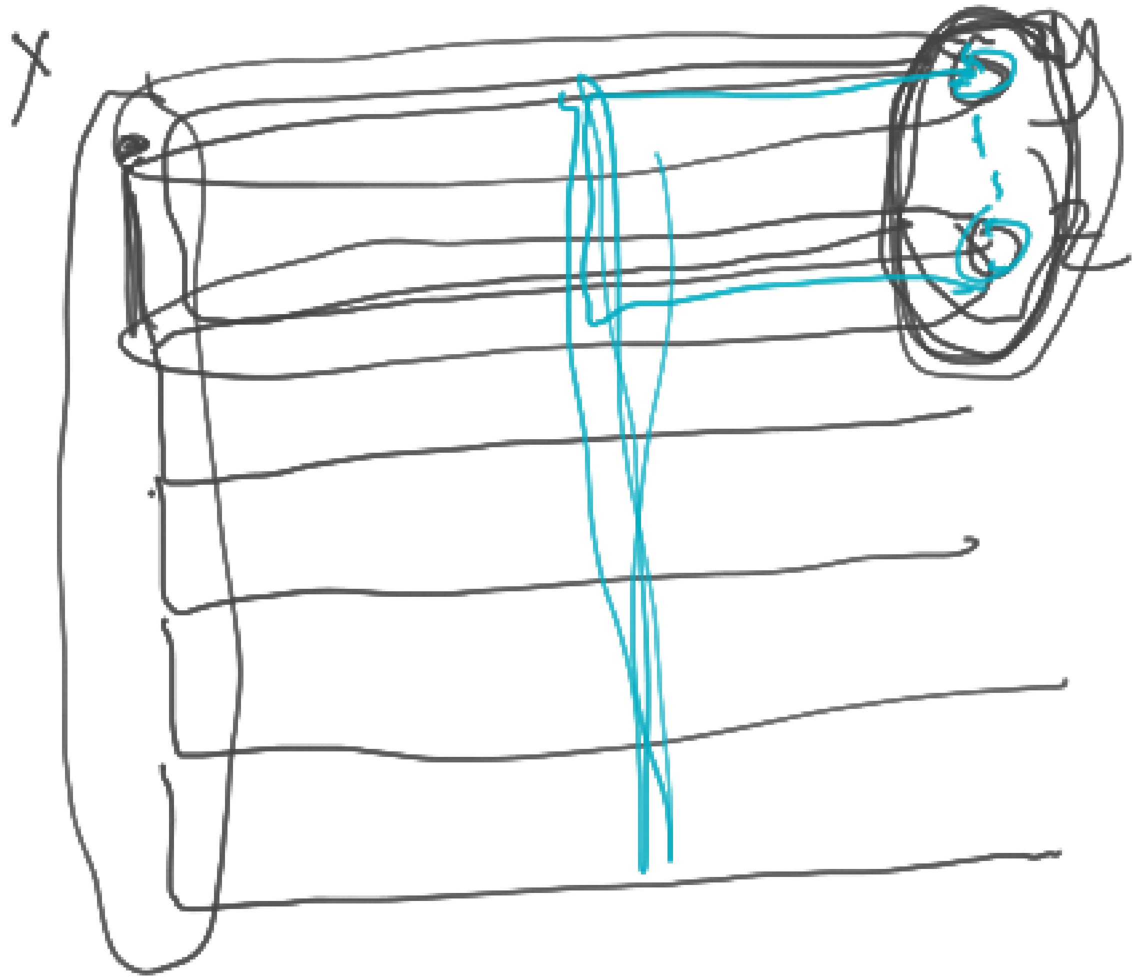
vertices is n.



$d(x,y) = 2\sqrt{n}$

No.

Diameter of G to be $\max_{x,y} d(x,y)$.



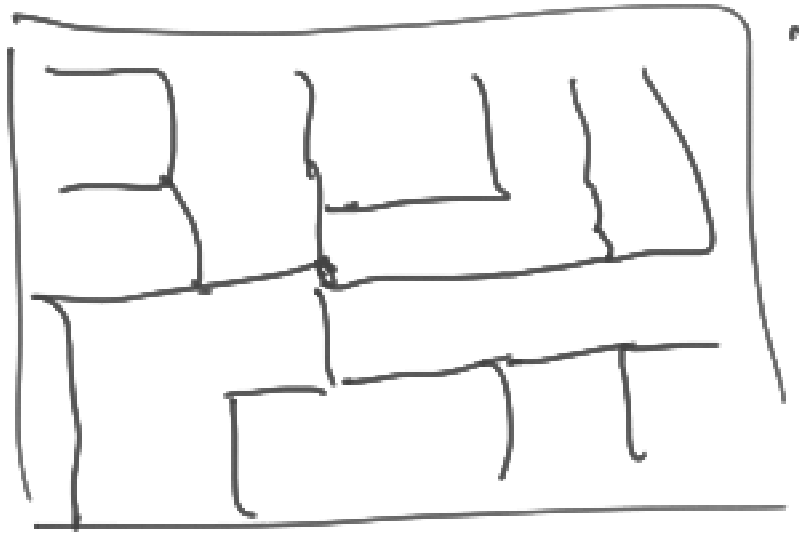
$$\mathrm{diameter}(T) = O(\sqrt{n})$$

$$d_T(y, z) = 2\sqrt{n} \cancel{+} 1$$

$$d_G(y, z) = \underline{1}$$

For grid graph: shortest path tree from a random vertex

random



Def (stretch). G, spanning tree T.

G is unweighted. is connected.

Stretch(u, v) = length between $\cancel{x}$ u and $\cancel{y}$ v in T.

$\boxed{(u,v)}$ is an edge of G

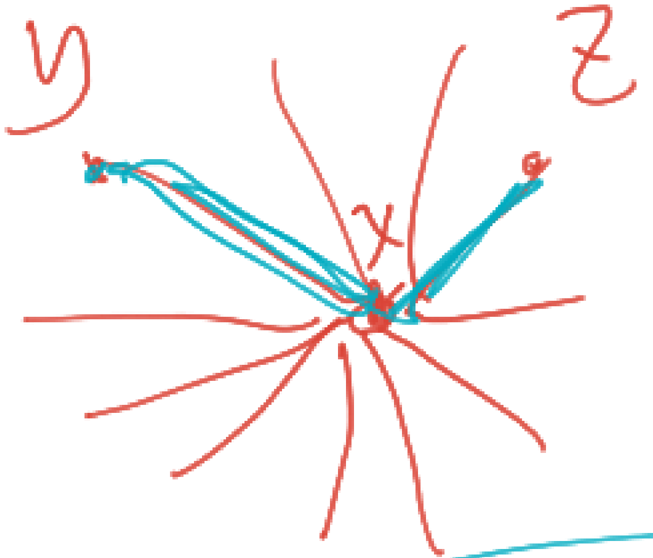
G is weighted

$$\text{Stretch}(u, v) = \frac{\text{Dist}_{T,w}(\cancel{x}\,u, \cancel{y}\,v)}{w(e)}.$$

Complete graph.

$m \le \text{Stretch}(G) \le 2 \cdot m$

\# edges



$\text{Stretch}(x, y) = 2$

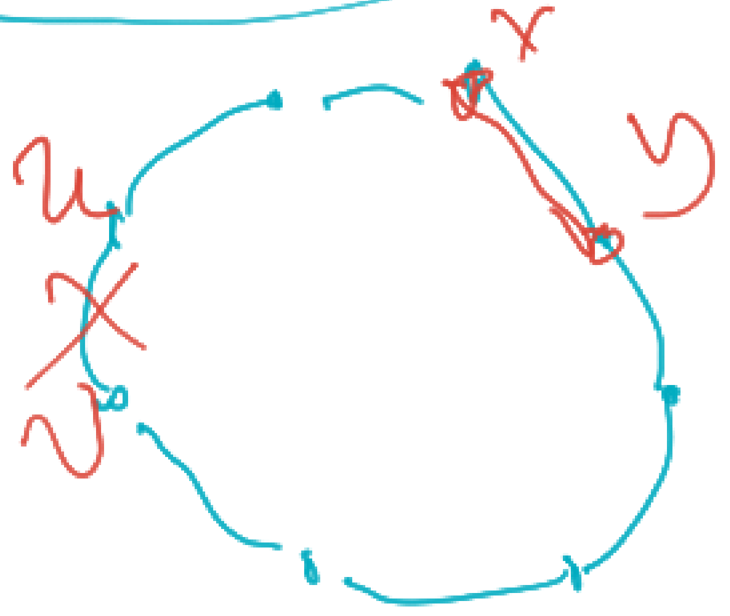
$\text{Stretch}(y, z) = 2.$

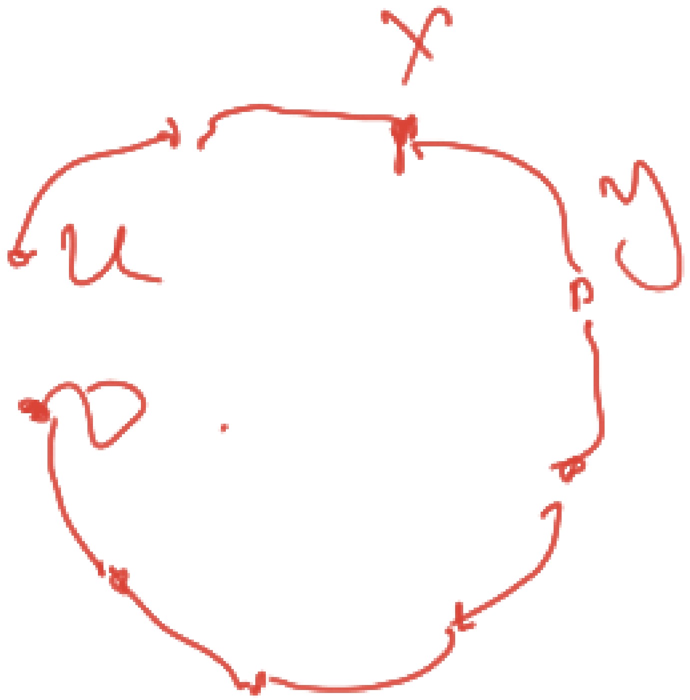
n vertices

$\binom{n}{2}$ edges.

$m \le \text{Stretch}(G) \approx 2m.$

$\text{Stretch}(x, y) = ?$





$\text{Stretch}(u, v) = n-1$

Not all the edges have good stretch.

Def total stretch: $G=(V,E)$

$$\text{Stretch}_T(G) = \sum_{e \in E} \text{Stretch}_T(e).$$

Goal: Construct spanning tree ~~has~~ with small stretch ideally $\cancel{O}(m)$.

Best $O(m \cdot \log n \cdot \log\log n)$

Polynomial Algorithm.

$\exists$ graph Stretch $\geq \Omega(m \cdot \log n)$.

Next Lecture

$O(m \cdot 2^{\sqrt{\log n}})$.

$\leq m^{1.01}$