Breaking Quadratic Time for **Small Vertex Connectivity**

Danupon Nanongkai
KTH

Thatchaphol Saranurak
TTIC

Sorrachai Yingchareonthawornchai
Michigan State U → Aalto University

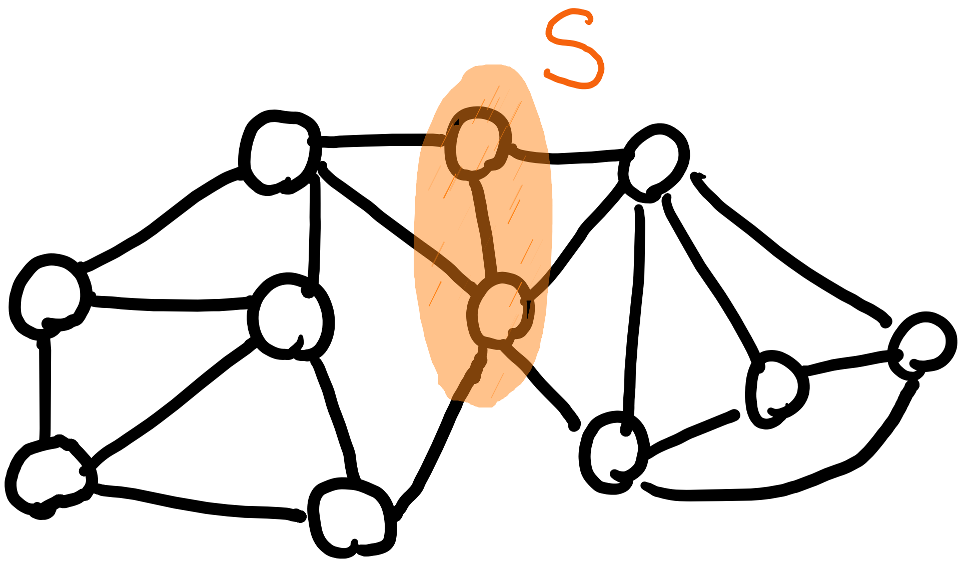
TCS+
17 April 2019

Definition: k-Connectivity

A graph G is ***k*-connected** iff

Cannot disconnect G by removing less than k nodes

($\forall S\ |S| < k$, then $G[V - S]$ has only one connected component)



G is 2-connected

G is not 3-connected

Given a graph $G = (V, E)$ with n nodes and m edges

Answer: is G k-connected?

*Hide log f

History (when k is small)

eference		
rivial	m	$k = 1$
arjan’72	m	$k = 2$
opcroft Tarjan’73	m	$k = 3$
onjecture: Aho, Hopcroft and Ullman’74	m	For any k
anevsky Ramachandran’91	n^2	$k = 4$
Our work	$\boldsymbol{k^2 m}$	

Our result

Prove the conjecture by [Aho, Hopcroft and Ullman'74] for $k = O(1)$ up to log factors

Break the 50-year-old bound of $O(n^2)$ since Kleitman'69

Fastest when $4 \leq k \leq n^{(\omega-1)/3} = n^{0.456}$

Other results

In directed graphs, need the same $\tilde{O}(mk^2)$ time
(and better bounds in dense graphs)

Comparison:
Directed k-**edge** connect
Best known: $O(mk)$
by [Gabow FOCS'00]

$(1+\epsilon)$-approximation in $\tilde{O}(mk/\epsilon)$ time
in both undirected and directed graphs
(and better bounds in dense graphs)

If I have a time machine…

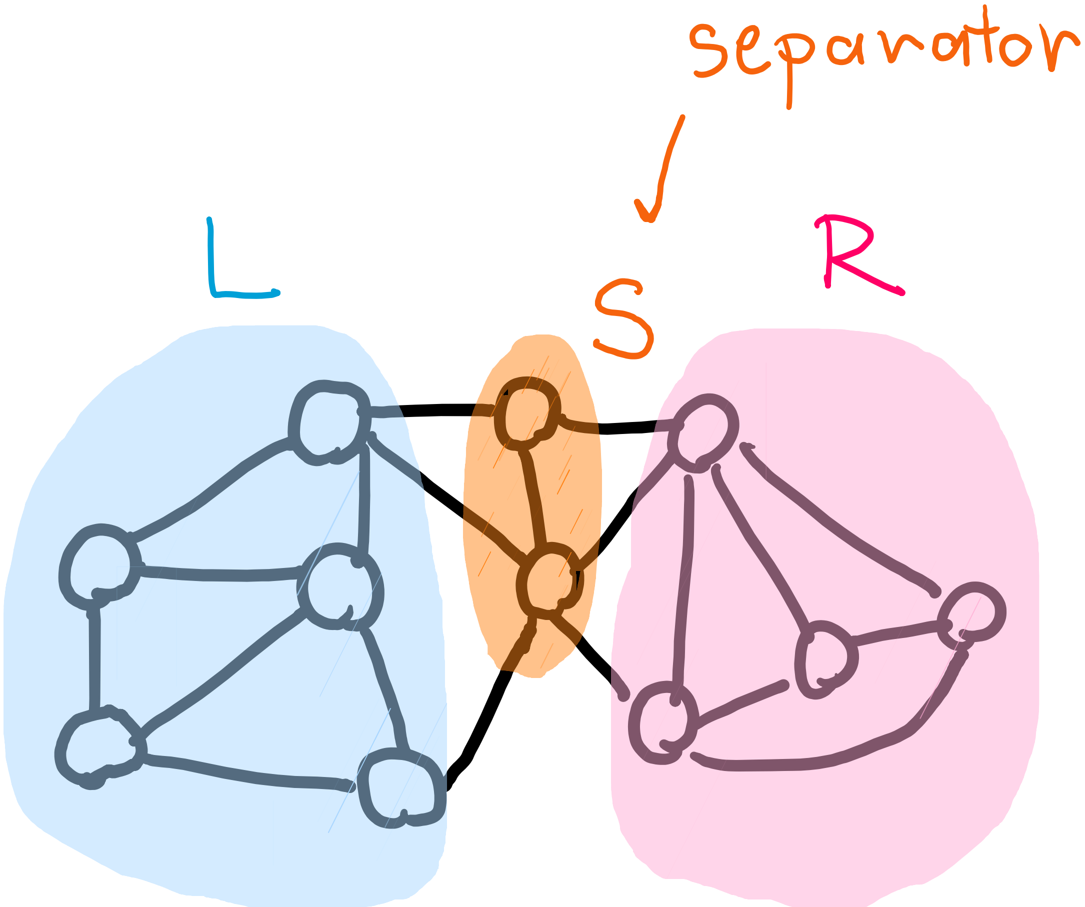
To people in **90**'s:
"try to solve **locally**"

To people in **70**'s:
"try to solve **locally** + use randomization"

Our algorithms could have been found long time ago
using basic techniques from the 70's

Part 0: Definitions

Definition: Vertex Cut (L,S,R)

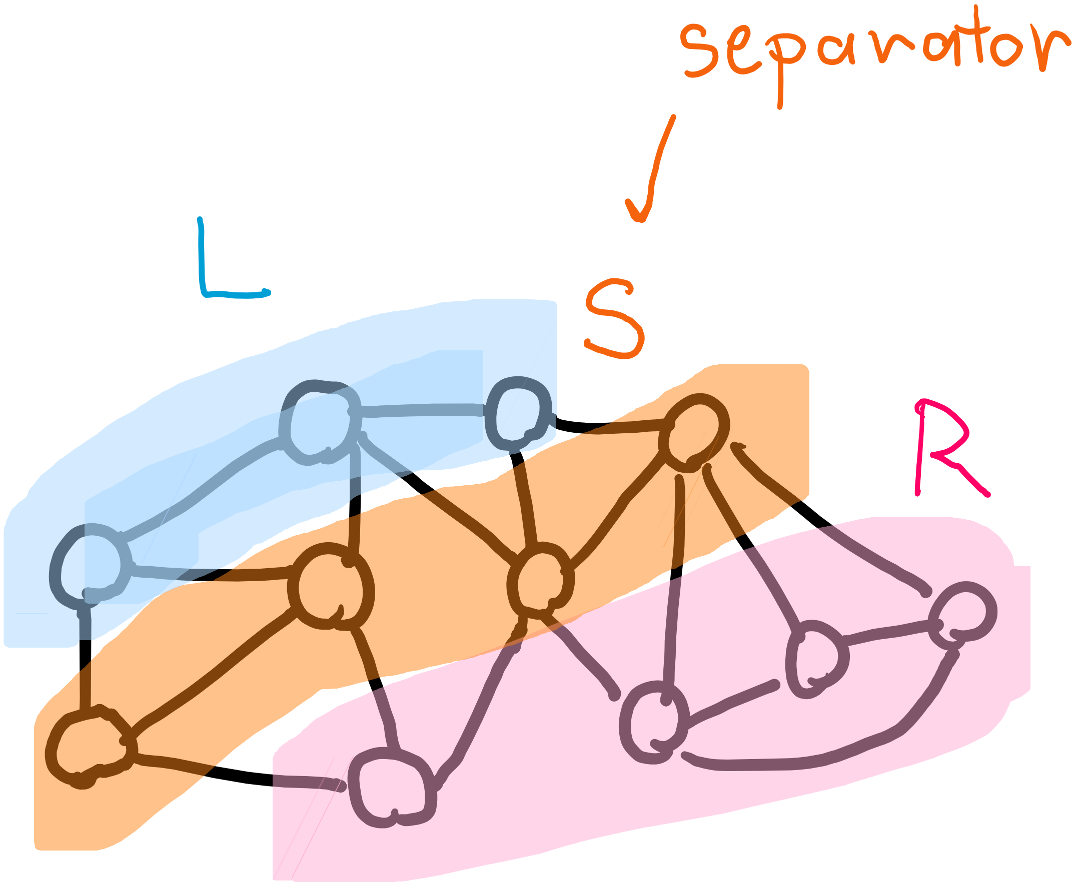


(L, S, R) is a **vertex cut**

- L, S, R partition V and $L, R \neq$
- No edge between L and R
  i.e. neighbors $N(L) = S = N($

Size of $(L, S, R) = |S|$

<u>Definition:</u> Vertex Cut (L,S,R)



(L, S, R) is a **vertex cut**

- L, S, R partition V and $L, R \neq$
- No edge between L and R
  i.e. neighbors $N(L) = S = N($

Size of $(L, S, R) = |S|$

Definition: s-t Connectivity

$\boldsymbol{\kappa(s,t)}$**:** size of min cut (L, S, R) where $s \in L$ and $t \in R$

Can check $\kappa(s,t) \geq k$ in $O(mk)$ time by Ford-Fulkerson

Note: G is k-connected iff $\kappa(s,t) \geq k$ for all $s, t \in$

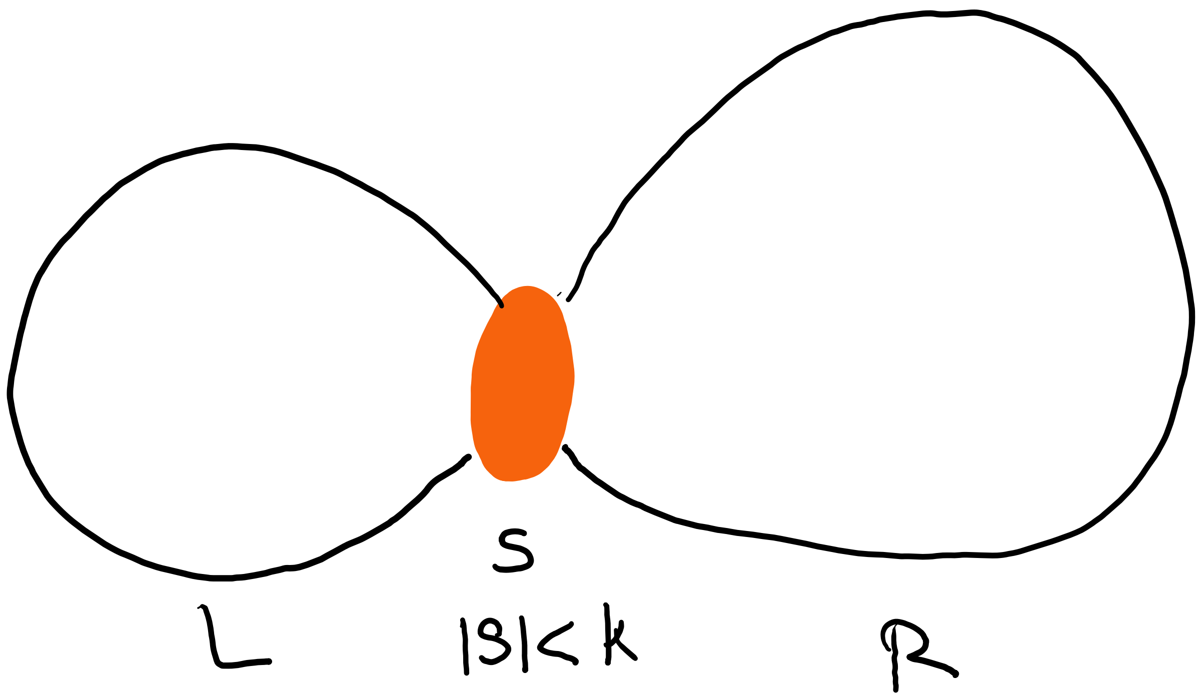
Part 1: The Framework

Suppose G is not k-connected

So, there is a cut (L, S, R) where $|S| < k$

Assume $\mathrm{vol}(L) \leq \mathrm{vol}(R)$ where $\mathrm{vol}(X) = \sum_{u \in X} \deg u$

Goal: w.h.p. find some cut (L', S', R') where $|S'| < k$



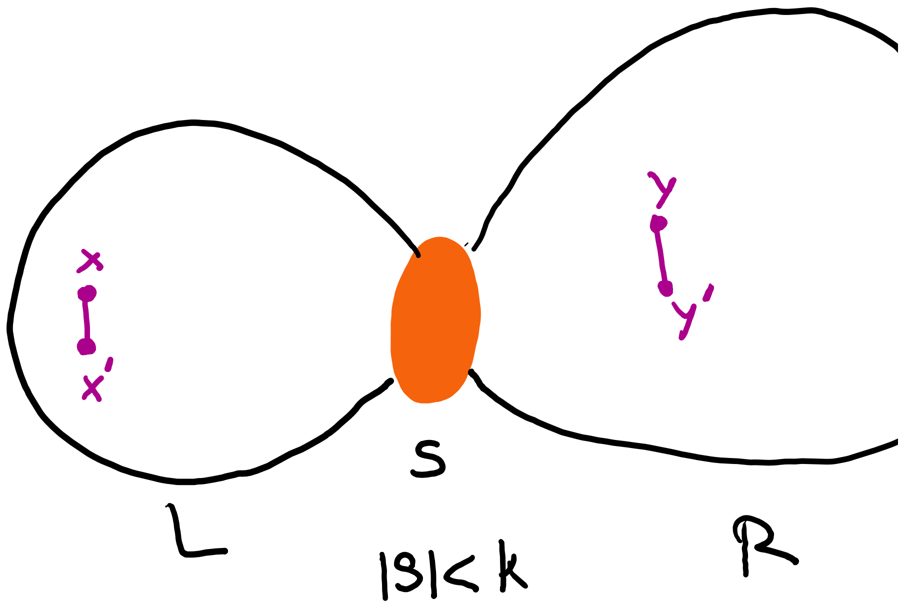
Case 1: Balanced $\mathbf{vol}(L) \geq \Omega(\frac{m}{k})$

Observe: If sample $\tilde{O}(k)$ pairs of edges $e = (x, x')$ and $f = (y, y')$, then, w.h.p., exists a sampled pair (e, f) where $x \in L$ and $y \in R$

Alg: For each sampled (e, f), if $\kappa(x, y) < k$, return a cut

must obtain a cut once w.h.p.

Time: $\tilde{O}(k) \times O(mk) = \tilde{O}(mk^2)$

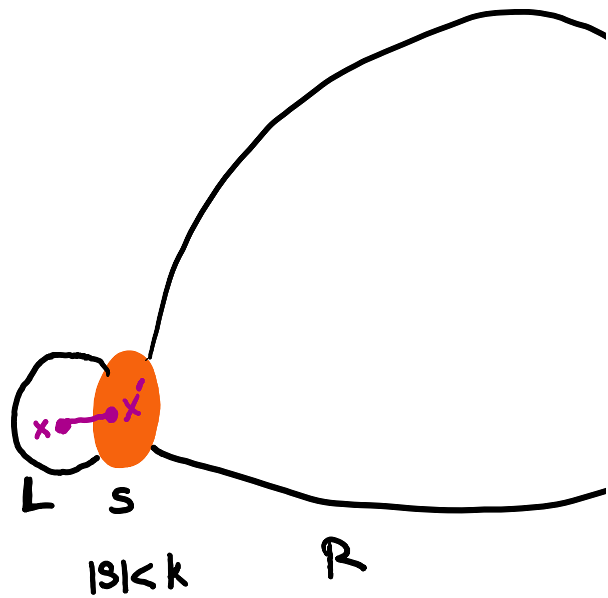


Case 2: Unbalanced $\mathrm{vol}(L) \le \frac{m}{k}$

Suppose $\mathrm{vol}(L) \in [2^{i-1}, 2^i]$.

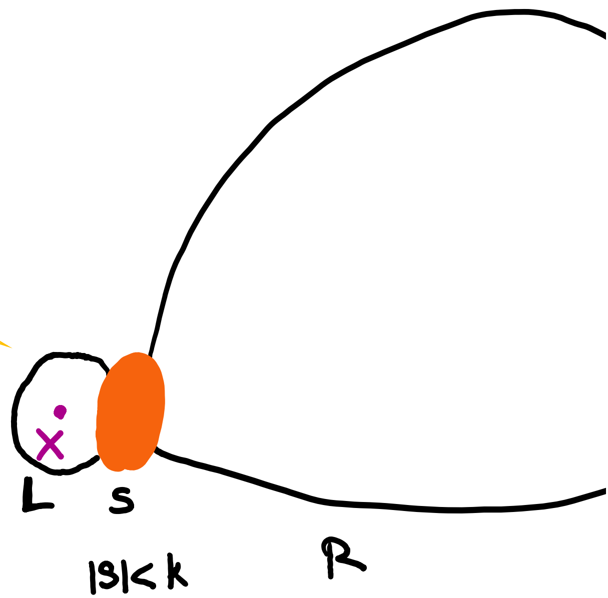
Observe: If sample $\tilde{O}(m/2^i)$ edges $e = (x, x')$
then, w.h.p., exists a sampled edge e where $x \in L$

Want: find the separator S "near" x in $\sim O(2^i)$ time
Cannot spend linear time per sample



<u>The key tool:</u> Local Vertex Connectivity

Informal: given a seed node x,
Find $L \ni x$ of small volume
and has cut-size less than k
in sub-linear time



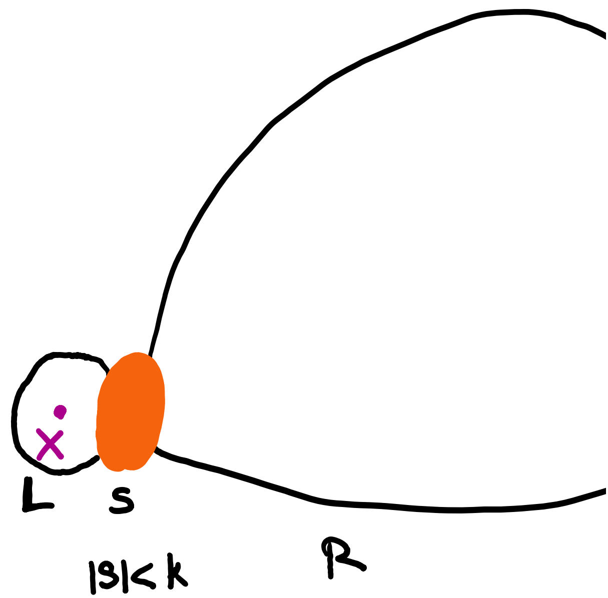
The key tool: Local Vertex Connectivity

Input: $\mathrm{Local}(x, \nu, k)$ where x is a node (and $\nu \leq m/k$)

Output: either

- Declare that no $L \ni x$ where $|N(L)| < k$ and $\mathrm{vol}(L) \leq \nu$
- Return $L \ni x$ where $|N(L)| < k$

We can do this in time: $\widetilde{\boldsymbol{O}}(\boldsymbol{\nu k^2})$
No dependency on n!



Case 2: Unbalanced $\mathrm{vol}(L) \leq \frac{m}{k}$

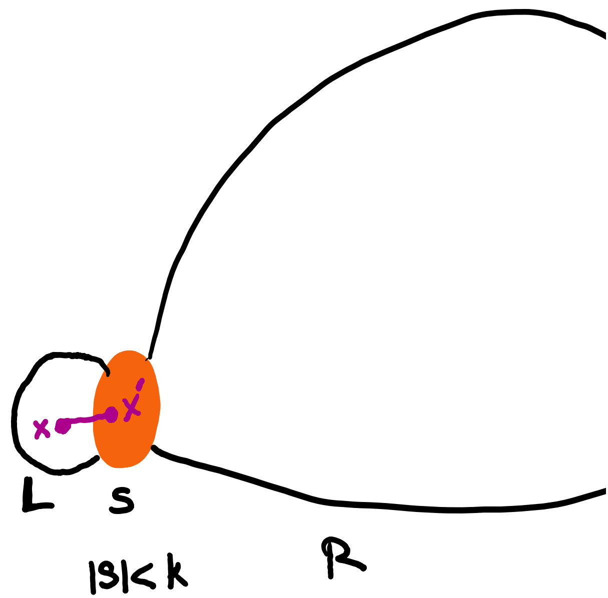
Suppose $\mathrm{vol}(L) \in [2^{i-1}, 2^i]$.

Observe: If sample $\tilde{O}(m/2^i)$ edges $e = (x, x')$ then, w.h.p., exists a sampled edge e where $x \in L$

Alg: For each sampled e, run $\mathrm{Local}(x, 2^i, k)$

must return a cut once w.h.p.

Time: $\tilde{O}(m/2^i) \times \tilde{O}(2^i k^2) = \tilde{O}(mk^2)$



The $\tilde{O}(mk^2)$-time Algorithm

1. Sample $\tilde{O}(k)$ pairs of edges $e = (x, x')$ and $f = (y, y')$
2. For each sampled (e, f), return a cut if $\kappa(x, y) < k$

Find a cut w.h.p. w
$\mathrm{vol}(L) \geq \Omega(\frac{m}{k}$
Time: $\tilde{O}(k) \times O(r$

3. For $i = 1, \ldots, \log \frac{m}{k}$
   1. Sample $\tilde{O}(m/2^i)$ edges $e = (x, x')$
   2. For each sampled e, run $\mathrm{Local}(x, 2^i, k)$

Find a cut w.h.p. when
$\mathrm{vol}(L) \approx 2^i \leq \frac{m}{k}$
Time: $\tilde{O}(\frac{m}{2^i}) \times \tilde{O}(2^i k^2)$

4. If did not find a cut, declare G is k-connected

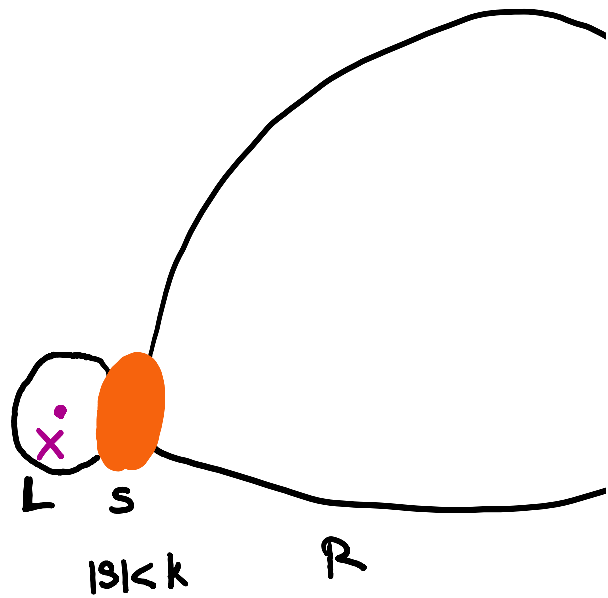
Part 2: Local Vertex Connectivity

Definition: Local Vertex Connectivity

Input: $\mathrm{Local}(x, \nu, k)$ where x is a node (and $\nu \leq m/k$)

Output: either

- Declare that no $L \ni x$ where $|N(L)| < k$ and $\mathrm{vol}(L) \leq \nu$
- Return $L \ni x$ where $|N(L)| < k$



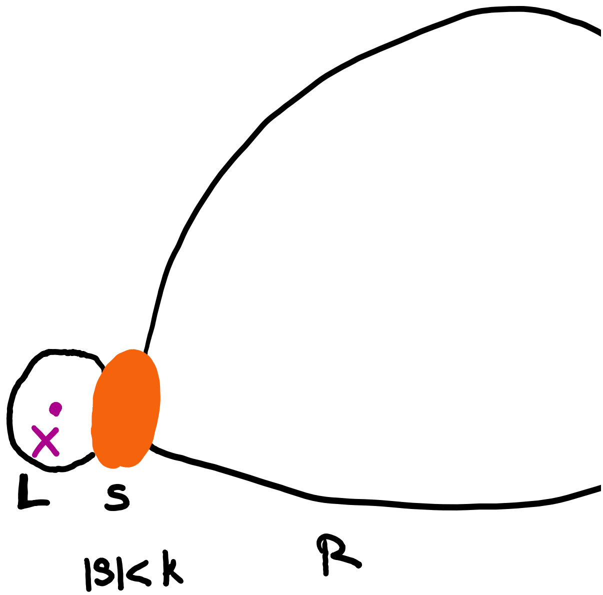
Part 2.1: Reducing to directed edge connectivity

Definition: Local Vertex Connectivity

Input: $\mathrm{Loc}al(x, \nu, k)$ where x is a node (and $\nu \leq m/k$)

Output: either

- Declare that no $L \ni x$ where $|N(L)| < k$ and $\mathrm{vol}(L) \leq \nu$
- Return $L \ni x$ where $|N(L)| < k$



Definition: Local Directed Edge Connectivity

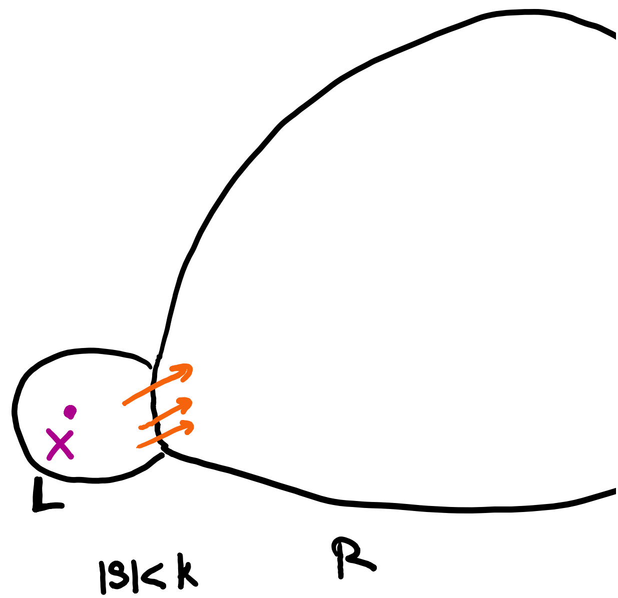
Input: $\mathrm{Local}(x, \nu, k)$ where x is a node (and $\nu \leq m/k$)

$\mathrm{vol}(L) = \sum_{u \in L} \deg_{out} u + \mathrm{de}$

Output: either

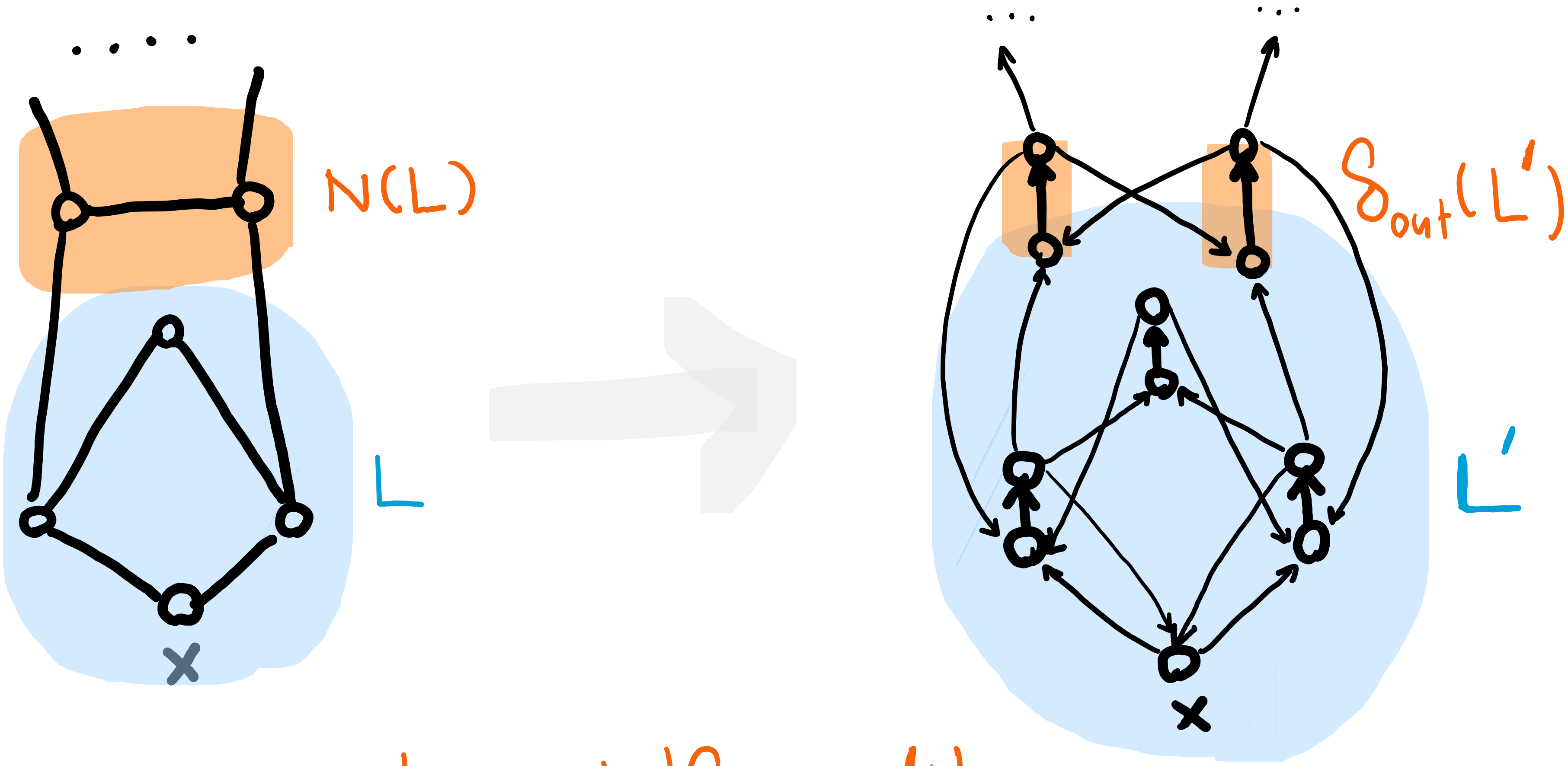
- Declare that no $L \ni x$ where $|\delta_{out}(L)| < k$ and $\mathrm{vol}(L) \leq \nu$
- Return $L \ni x$ where $|\delta_{out}(L)| < k$

$\delta_{out}(L) = E(L, V - L)$



Reducing from **vertex** to **directed edge** connectivity

Reducing from **vertex** to **directed edge** connectivity



$$|N(L)| = |\delta_{out}(L')|$$

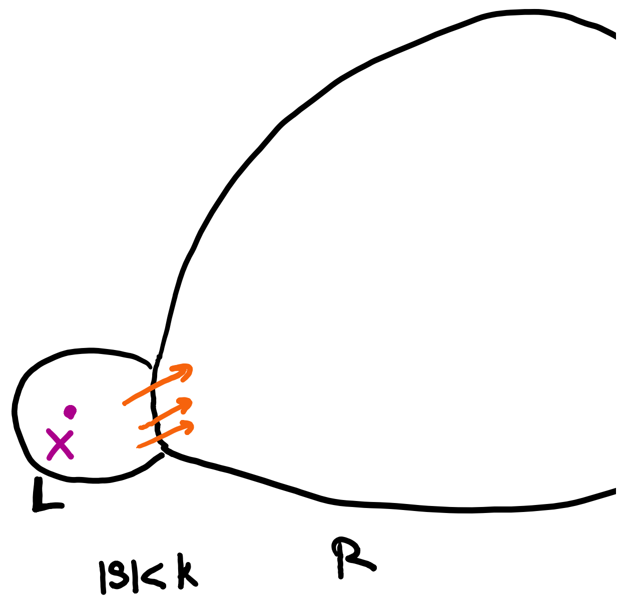
$$vol(L) = \Theta(vol(L'))$$

Definition: Local Directed Edge Connectivity

Input: $\mathrm{Local}(x, \nu, k)$ where x is a node (and $\nu \leq m/k$)

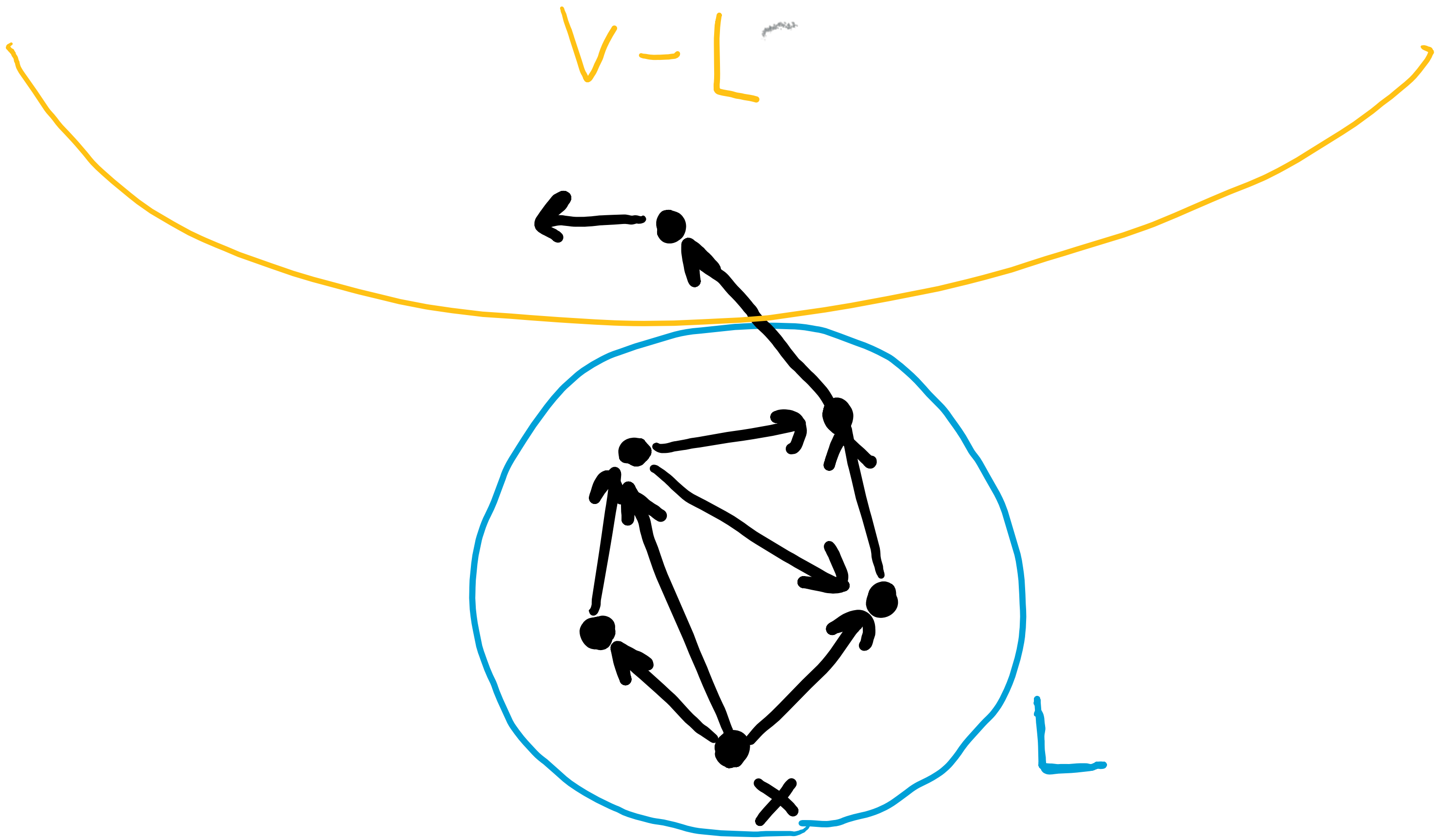
Output: either

- Declare that no $L \ni x$ where $|\delta_{out}(L)| < k$ and $\mathrm{vol}(L) \leq \nu$
- Return $L \ni x$ where $|\delta_{out}(L)| < k$



Part 2.2: Warm-up

Suppose $|\delta_{out}(L)| = 1$ and $|\delta_{in}(L)| = 0$



DFS from x

Explore exactly $\nu+1$ edges.

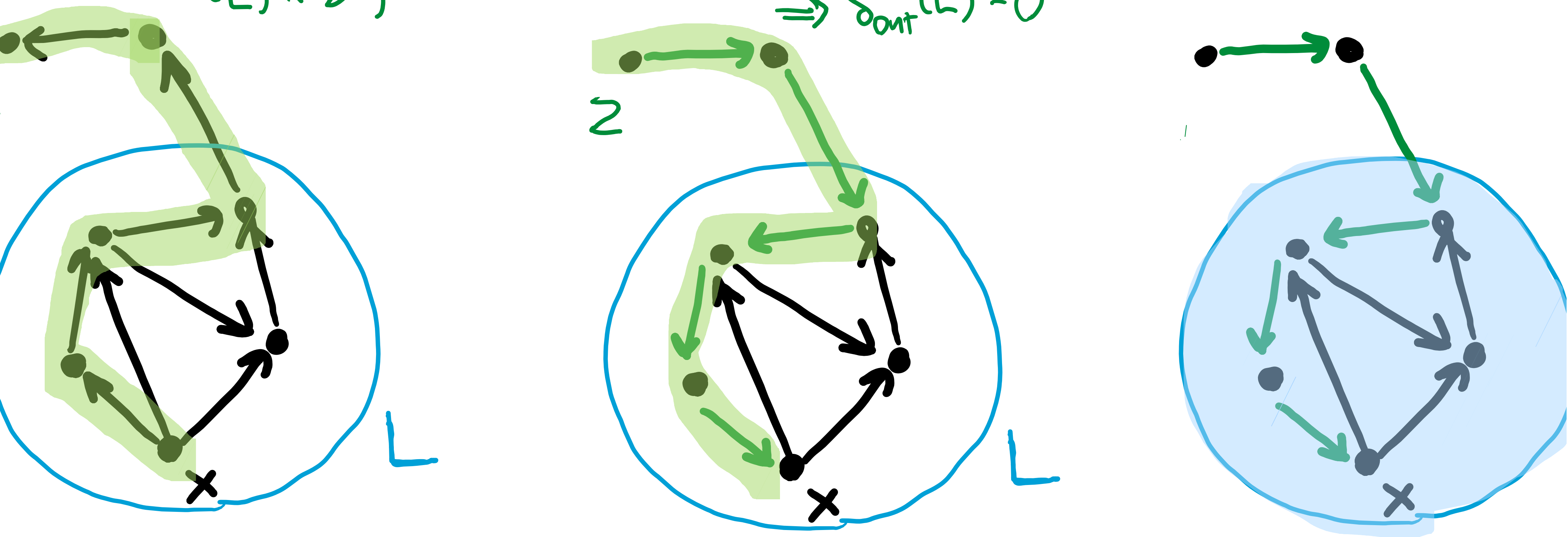
$\Longrightarrow$ must go out of L!

$(vol(L) < \nu)$

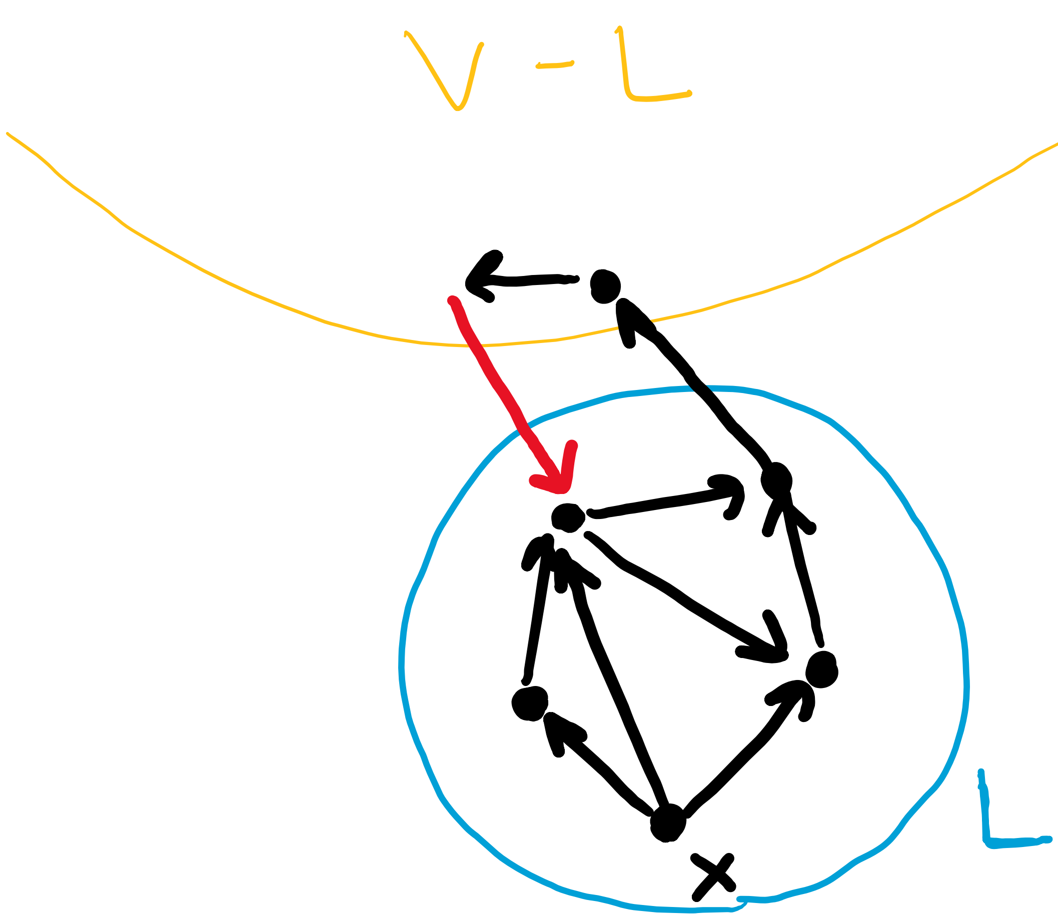
Reversing path P_{xz} from $x \to z$

$\Longrightarrow \delta_{out}(L) = 0$

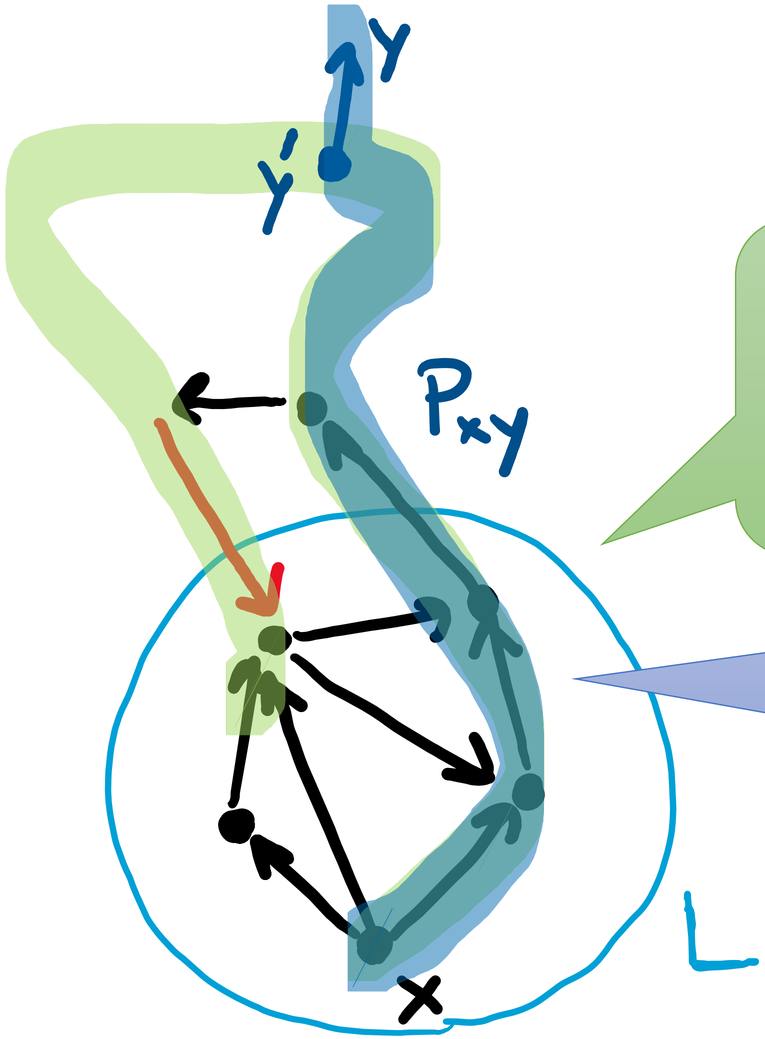
Next DFS will get stuck and obtain L



What if $|\delta_{in}(L)| > 0$?



Idea: Sampling an Explored Edge



- Sample an explored edge (y', y).
- Most explored edges are outside L, so $\boldsymbol{y}$ **should be outside** $\boldsymbol{L}$**.**

- **Reversing path** $\boldsymbol{P_{xy}}$ from x to y will reduce $\delta_{out}(L)$.

Part 2.3: The Algorithm

1. Time: $\boldsymbol{k} \times \boldsymbol{O(\nu k)} = O(\nu k^2)$

Repeat $\boldsymbol{k}$ times.

1. Grow DFS tree T from x and explore exactly $\boldsymbol{k \cdot \nu}$ edges
2. If get stuck, found the cut L'. **Terminate**
3. Sample an explored edges (y', y).
4. Reverse the path P_{xy} in T

Terminate with no cut.

2. Soundness: If L' is return then $\delta_{out}(L') < k$

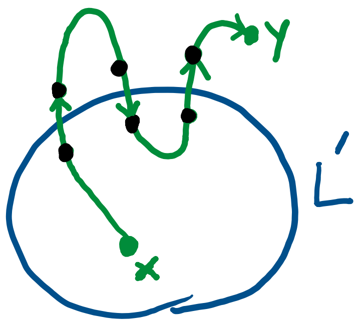
3. Completeness: If no cut returned, then w.p. $> \mathbf{1/10}$ there is no $L \ni x$, where $\text{vol}(L) < \nu$ and $\delta_{out}(L) < k$

Can repeat $\tilde{O}(1)$ times to get high probability

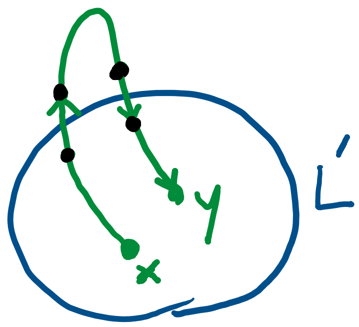
Soundness

Fix any $L' \ni x$. Suppose we reverse P_{xy}.

If $y \notin L'$, $\delta_{out}(L') \leftarrow \delta_{out}(L') - 1$



If $y \in L'$, $\delta_{out}(L')$ stays the same



At the end $\delta_{out}(L') = 0$ and there were $< k$ path-reversions

$\Rightarrow$

$\delta_{out}(L') < k$ initially

Part 3: Recap

The $\tilde{O}(mk^2)$-time Algorithm

1. Sample $\tilde{O}(k)$ pairs of edges $e = (x, x')$ and $f = (y, y')$
2. For each sampled (e, f), return a cut if $\kappa(x, y) < k$

Find a cut w.h.p. w

$\mathrm{vol}(L) \geq \Omega(\frac{m}{k}$

Time: $\tilde{O}(k) \times O(r$

3. For $i = 1, \ldots, \log \frac{m}{k}$
   1. Sample $\tilde{O}(m/2^i)$ edges $e = (x, x')$
   2. For each sampled e, run $\mathrm{Local}(x, 2^i, k)$

Find a cut w.h.p. when

$\mathrm{vol}(L) \approx 2^i \leq \frac{m}{k}$

Time: $\tilde{O}(\frac{m}{2^i}) \times \tilde{O}(2^i k^2)$

4. If did not find a cut, declare G is k-connected