

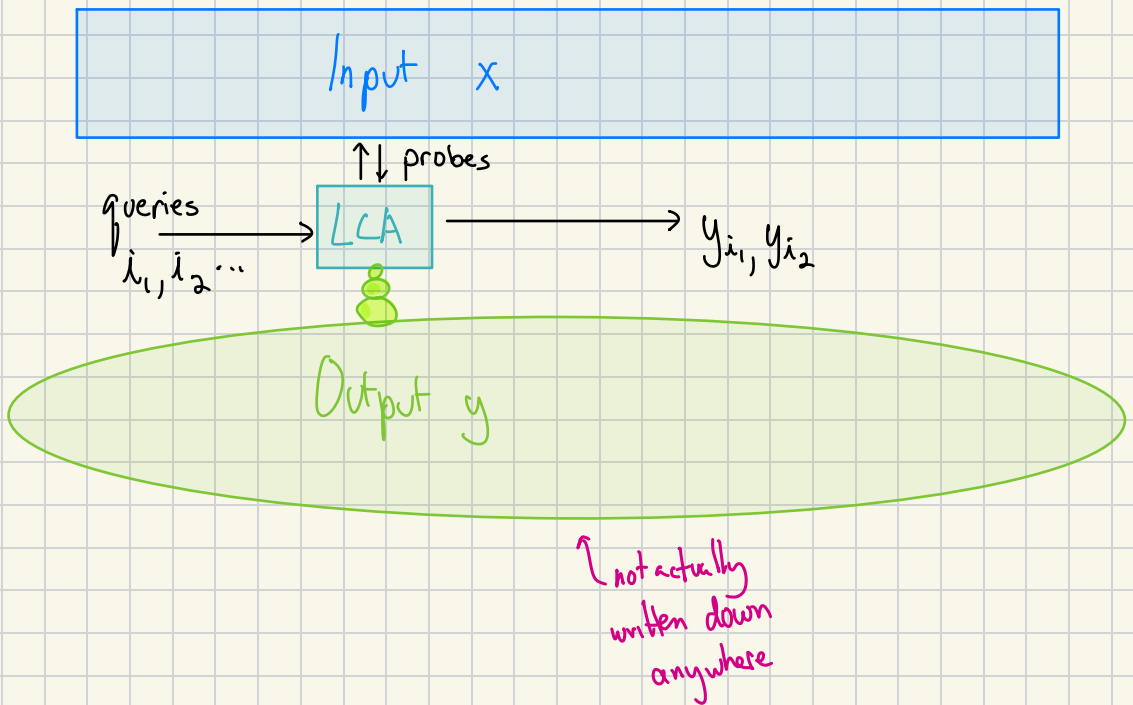
Local Computation Algorithms:

- the model
- Maximal Independent Set

Local Computation Algorithms (LCAs)

when input and output large,
do we need to construct full output?
" " " " read " input?

LCAs provide fast query access to output



For example:

Maximal Independent Set:

$X =$ description of graph
 $Y = (y_1, \dots, y_n)$ s.t. $y_i = \begin{cases} 1 & \text{if node } i \text{ in MIS} \\ 0 & \text{o.w.} \end{cases}$

def $U \subseteq V$ is a "maximal independent set" (MIS) if

(1) $\forall u_1, u_2 \in U \quad (u_1, u_2) \notin E$ "independent"

(2) $\nexists w \in V \setminus U$ s.t. $U \cup \{w\}$ is independent
"maximal"

G has max degree Δ

Queries: is node u in MIS? $\leftarrow$ do we need to compute whole MIS in order to answer?

Problem: lots of MIS's possible!
every u is in some MIS
Can we always answer "yes"?

How to define LCA?

more than one "legal" output?

depends on query order?

need to remember past answers?

would like answer to be "No" for above

def A is "Local computation algorithm" (LCA)
for a problem Π if

- given:
 - probe access to input X
 - random bits r
 - local memory
- answers queries to bits/words of

~~$y = \Pi(x)$~~
 $y \in \Pi(x)$

what if more than one soln?

maybe $y = \Pi_r(x)$?
← determined by random bits

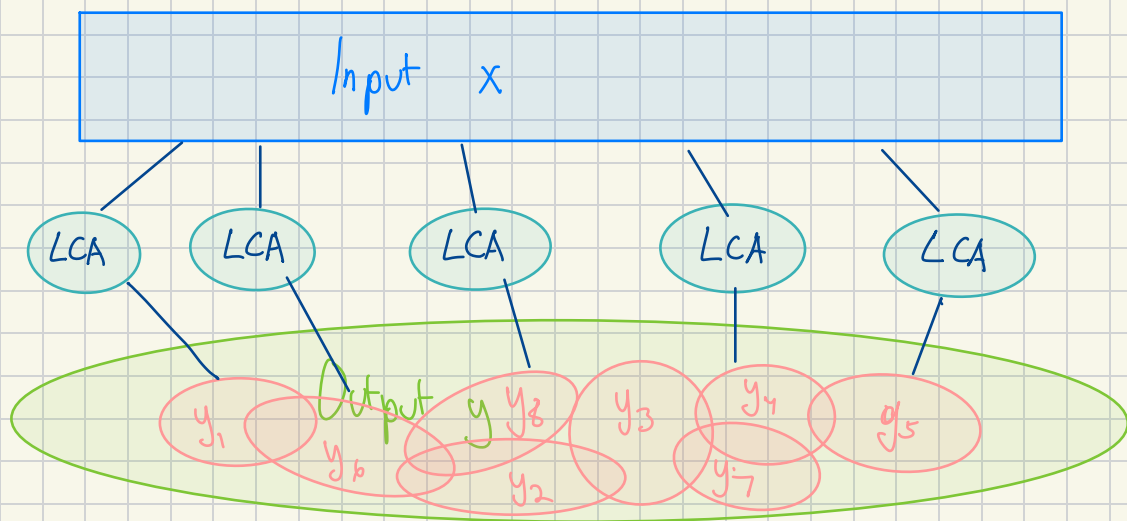
- memory of LCA "wiped" clean between queries (keep the random bits)

- $\forall$ queries i ,
let $y_i \leftarrow$ output of LCA on i
- note: can't depend on other queries since "wiped"

$\forall x$, whp over r ,
 $\forall$ queries y

then y_i 's must be consistent with a legal solution for input x . (e.g. $\in \Pi(x)$)

Can we provide "illusion" of fully constructed output?



Equivalent model:

- many copies, each gets only one question
 - initially share (short?) random string
 - afterwards compute independently
- sublinear time, space, randomness

Maximal Independent Set:

$X = \text{description of graph}$
 $Y = (y_1, \dots, y_n)$ s.t. $y_i = \begin{cases} 1 & \text{if node } i \text{ in MIS} \\ 0 & \text{o.w.} \end{cases}$

def $U \subseteq V$ is a "maximal independent set" (MIS) if

(1) $\forall u_1, u_2 \in U \quad (u_1, u_2) \notin E$ "independent"

(2) $\nexists w \in V \setminus U$ s.t. $U \cup \{w\}$ is independent "maximal"

G has max degree Δ

note MIS can be solved via greedy

but maximum independent set is NP-complete

Plan • show distributed alg for MIS in k rounds
• Convert to $\Delta^{O(k)}$ sublinear query/time algorithm
(using reduction from last time)

Distributed Algorithm for MIS:

"Luby's Algorithm" (actually a variant...)

- $MIS \leftarrow \emptyset$
 - all nodes set to "live"
 - repeat k times:
 - (in parallel)
 - $\forall$ nodes v , color self "red" with prob $\frac{1}{2\Delta}$ else "blue"
 - send color to all nbrs
 - if v colors self red, & no other nbr of v colors self red, then
 - add v to MIS
 - remove v & all nbrs from graph (set to "inactive")
 - all nodes output status ("live", in MIS, not in MIS)
- * (For purposes of analysis, continue to select colors after removed but don't send to nbrs)

Thm $\Pr[\# \text{ rounds til graph empty} \geq 8\Delta \log n] \leq \frac{1}{n}$

Corr $E[\# \text{ rounds}]$ is $O(\Delta \log n)$ $\leftarrow$ can improve!

Main Lemma: For live v , $\Pr[v \text{ added to MLS in a round}] \geq \frac{1}{4\Delta}$

Proof:

$$\Pr[v \text{ colors self red}] = \frac{1}{2\Delta}$$

$$\begin{aligned} \Pr[\text{any } w \in \text{Nbr}(v) \text{ colors self red}] &\leq \sum_{w \in \text{Nbr}(v)} \frac{1}{2\Delta} \quad (\text{union bound}) \\ &\leq \frac{1}{2} \quad (\text{degree bound}) \end{aligned}$$

$$\begin{aligned} \therefore \Pr[v \text{ colors self red} \wedge \text{no other nbr colors self red}] \\ \geq \frac{1}{2\Delta} \cdot \left(1 - \frac{1}{2}\right) = \frac{1}{4\Delta} \quad (\text{independence}) \end{aligned}$$

■

$$\begin{aligned} \text{Lemma} \Rightarrow \Pr[v \text{ live after } \underbrace{4 \cdot k' \cdot \Delta}_{=K} \text{ rounds}] &\leq \left(1 - \frac{1}{4\Delta}\right)^{4\Delta \cdot k'} \\ &\leq e^{-k'} \end{aligned}$$

Setting k :

$$\begin{aligned} \text{if } k = \Theta(\Delta \log n), \Pr[v \text{ live at end}] &\leq e^{-\Theta(\log n)} \\ &= \frac{1}{n^c} \end{aligned}$$

(can do better)

Problem: when sequentially simulate $\Theta(\Delta \log n)$ rounds,
 $\approx n^{\Delta \log \Delta}$ need $\Delta^{\Theta(\log n)}$ complexity ... not sublinear in n 😞

Idea

- define "Luby-status":
run Luby for $k = \Theta(\Delta \log \Delta)$ rounds

at end, each $v \in V$ is one of

live
in MIS
not in MIS

← set self to red
& no nbrs did
← taken out by nbr in MIS

- use [Parnas-Ron] (distributed $\rightarrow$ sequential) reduction
to simulate v 's view of Luby-status in sequential
manner:
 $\Delta^k = \Delta^{\Theta(\Delta \log \Delta)}$
↑ degree ← rounds queries

- if v is in/not in then done
else v is alive ← what do we do now?

Questions:

- what is probability that r still alive?

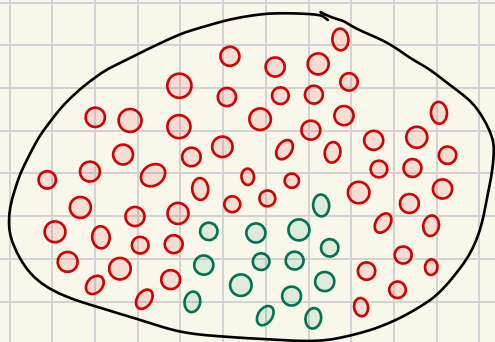
$$\begin{aligned} \Pr[r \text{ inactive}] &\leq \Pr[\text{survives } \Theta(\log \Delta) \text{ rounds}] \\ &\leq e^{-\Theta(\Delta \log \Delta)} \leq \frac{1}{\Delta^c} \end{aligned}$$

for any c & for proper choice of constant in # rounds

Very few stay active!

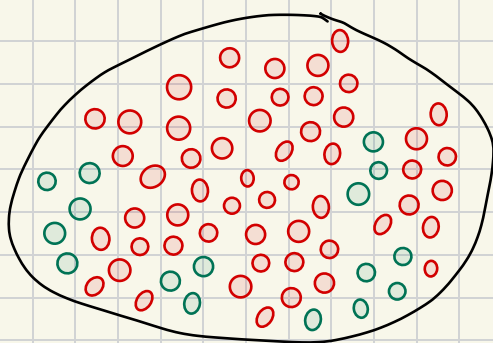
- how much work to "finish off" r ?

Key insight distribution of live nodes!



big clumps of surviving nodes?

NO!



surviving nodes in small connected components

relies on degree bound of graph
 $\Rightarrow$ survival of components $\approx$ independent

LCA for MIS(v):

- run sequential version of LubyStatus(v)
- if it is "in"/"out" output answer & halt
- else, (1) do BFS, run LubyStatus(w) at each newly visited node, to find v's connected component of **live** nodes

Sequential
runtime
 $\Delta^{O(\Delta \log \Delta)}$

$\Delta^{O(\Delta \log \Delta)}$ $\times$ size of
Component

(2) compute lexicographically first MIS M' for
that connected component

(note that nbors outside connected component
are all "out" of MIS)

use IDs $\rightarrow$
to determine
lexicographical
order

size of
Component

(3) Output whether v "in"/"out" of M'

What is size of
Component?

Bounding size of connected components:

Claim After $O(\Delta \log \Delta)$ rounds, connected components of survivors of size $\leq O(\text{poly}(\Delta) \cdot \log n)$

$\Rightarrow$ can find whole component via BFS
"brute force"

total runtime is $O(\underbrace{\Delta^{O(\Delta \log \Delta)}}_{O(\text{poly}(\Delta) \log n)} \cdot \log n)$
 $O(\text{poly}(\Delta) \log n)$ calls to
"Luby-status"

Main difficulty:

- survival of v + nbrs not independent in a round
- not independent of previous rounds

will show: (1) any big set has lots of independent events
(2) lots of independent events unlikely to happen simultaneously

Bounding survivors: independence between rounds
 some interesting variables...

$$A_v = \begin{cases} 1 & \text{if } v \text{ survives all rounds} \\ 0 & \text{o.w.} \end{cases}$$

$$B_v = \begin{cases} 1 & \text{if } \nexists \text{ round s.t. } v \text{ colors self +} \\ & \text{no } w \in N(v) \text{ colors self} \\ 0 & \text{o.w.} \end{cases}$$

$B_v = 0$

might

not mean that $v \in \text{MLS}$
 e.g. if v inactive due to nbr being put in MLS
 in earlier round

Note: $A_v = 1 \Rightarrow B_v = 1$

e.g. v survives $\Rightarrow \nexists$ round s.t. v colors self
 + no $w \in N(v)$ colors self

$$\Pr[B_v = 1] \leq \left(1 - \frac{1}{4\Delta}\right)^{C \cdot \Delta \log \Delta} \leq \frac{1}{8\Delta^3}$$

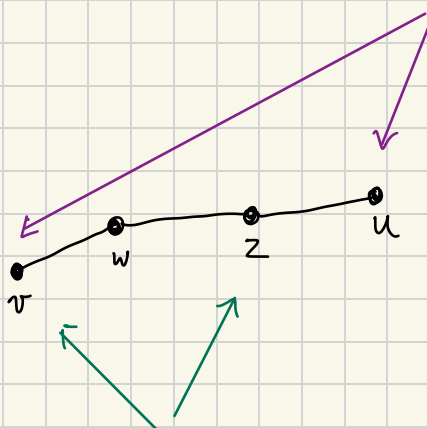
prob survive
one round

for $C \geq 20$

this is
where
we
need
X
in algorithm

these events
are
independent
between rounds!!

We care about the A_v 's, but the B_v 's
have nice independence properties



distance 3:

B_v depends on B_w

B_u depends on B_z

but $B_u + B_v$ are
independent!

distance 2:

$B_v + B_z$ depend

on w 's coins,

so not independent

$\deg \leq \Delta \Rightarrow$ each B_u depends on $\leq \Delta^2$
other B_w 's

Plan to bound size of connected component:

- any large ^{size $\geq w$} conn component has lots of nodes that are dist ≥ 3 so independent!
- these independent nodes are unlikely to simultaneously survive

BUT there are lots of large conn comps!!
do we need to union bound over all sets of size w ?

NO

Define new graph: the hope... nodes in new graph are independent?

Let $G^{(3)} \leftarrow$ graph with

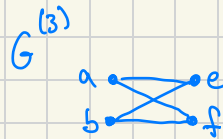
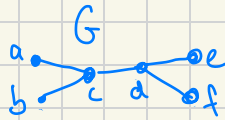
nodes $\sim B_v \quad \forall v \in G$

edges $\sim B_v + B_w$ distance $= 3$ in G

$$\deg(G^{(3)}) \leq \Delta^3$$

hope: represent independent events!!
(but not there yet)

What is $G^{(3)}$?



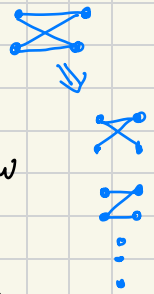
$G^{(3)}$ might not even be connected?

...

First,

Let's simplify conn. comp. to trees:

Observe: # conn comp in $G^{(3)}$ of size w
 $\leq$
size w subtrees in $G^{(3)}$



why? map each component C to
arbitrary spanning tree of C
mapping is 1-1 (but could have
many spanning trees per component)

great! we are good at counting trees!

Claim

For (live) connected component U in G ,
 $G^{(3)}$ contains tree with vertex set T
as subgraph +

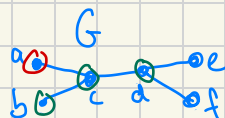
$$(1) |T| \geq \frac{|U|}{\Delta^2 + 1}$$

big!

$$(2) \text{dist}_G(u, v) \geq 3 \quad \forall u, v \in T \quad \left\{ \begin{array}{l} \text{so independent!!} \end{array} \right.$$

Proof

Pick T greedily



1. pick arbitrary $v \in U$

2. repeat until U empty

3. move v from U to T
remove all u with $\text{dist}_G(u, v) \leq 3$
from U

4. pick new node $v \in U$
s.t. $\text{dist}_G(u, v) = 3$ for
some $u \in T$

so $(u, v) \in G^{(3)}$

Note: for each v put in T , remove $\leq \Delta^2$ nodes from U
+ remove v
total $\leq \Delta^2 + 1$

How big are the remaining components?

Let $s \leftarrow \log 3N$

Let $\mathcal{T}_s \leftarrow \{ T \subseteq V \mid |T|=s, \text{ all } u, v \in T \text{ have } \text{dist} \geq 3 \}$
in G & T connected in $G^{(3)}$ }

$\Pr [\exists T \in \mathcal{T}_s \text{ s.t. all nodes in } T \text{ survive}]$

$$\leq \sum_{T \in \mathcal{T}_s} \Pr [\text{all nodes in } T \text{ survive}] \quad \text{union bound}$$

$$\leq |\mathcal{T}_s| \cdot \left(\frac{1}{8\Delta^3}\right)^s$$

$$\leq \left(N \cdot (4\Delta^3)^s\right) \cdot \left(\frac{1}{8\Delta^3}\right)^s \leq \frac{N}{2^s} = \frac{1}{3}$$

note that if we had union bndd over all subsets of nodes of size s , we would have $\sim N^s$

see next page for bound on #trees

(so, unlikely any size s tree survives)

$\Rightarrow$ with prob $\geq 2/3$ all surviving conn. comp. have size $\leq (\Delta^2 + 1) \cdot \log(3N)$
 $= O(\Delta^2 \log n)$

How many subtrees in a degree bounded graph?

Known Thm # non isomorphic trees on w nodes $\leq 4^w$

ignores names of nodes
(just shape)

Corr # size w subtrees in N -node graph of degree $\leq D$
is $\leq N \cdot 4^w \cdot D^w = N(4D)^w$

considers names of nodes

why?

- choose root in H
- choose size w tree shape from "known thm"
- choose placement in H via DFS sequence

#choices
 N

4^w

$\leq D$ choices for 1st child
 $\leq "$ " " 2nd step

⋮

total # choices: $N \cdot 4^w \cdot D^w$
 $= N(4D)^w$