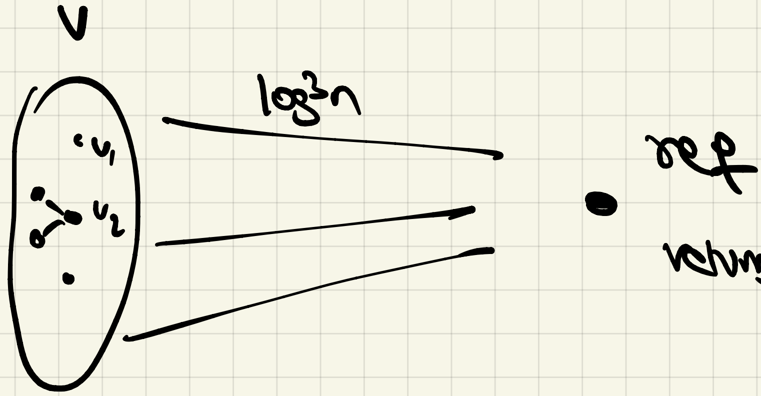
Sick algorithm for graph con

everyone sees

Shared random

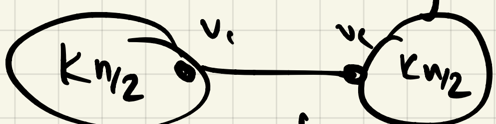


returning whether G is connected
if so ret spanning tree

→ each vertex only knows adjacent ones and sends over $\log^3 n$ info to ref who decides

Can't select info naively! (randomly)

e.g



→ v_1, v_2 very unlikely to send over (v_1, v_2) as edge info ∴ ref can't say if connected

Problem: a list of $2n+1$ numbers in $\{0, \dots, n^5\}$
promise: each number appears twice
number i appears once

find i

Soln: XOR all the numbers left with i

Alg for ref

$T = \emptyset$ → tree → coll of disj tree

Inv: T is a ~~Forest~~ $\subseteq E$

Repeat: (....) times → $\log n$ times → number of trees half

(in parallel) for each $T' \in T$ → $u \in V(T')$

(sequential) (*) Pick an edge $e = (u, v)$ $v \notin V(T')$

for each $e = (u,v)$ from (*) if u, v are disc in T

$T = T \cup \{u, v\}$

→ implement (*) using ~~wanted~~ info!

Implement following!

cut($S \subseteq V$): return (u,v) with $u \in S$ $v \notin S$

1st try

$sk(v)$ → sketch of a vertex's neighborhood

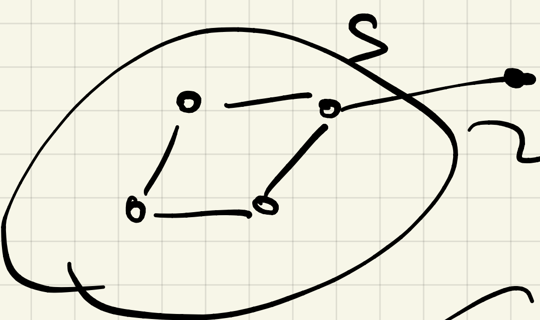
$= \bigoplus \langle e \rangle$ → encoding of the edge!

$e = (u,v) \in E$

$\langle (u,v) \rangle = \langle \min\{u,v\}, \max\{u,v\} \rangle$ → concatenate → left undir

→ $v = \langle \ldots \rangle$

S



→ goal we want to find edge leaving S

→ same as earlier

if (u,v) s.t. $u \in S, v \in S$

$\Rightarrow sk(u) = sk(v)$

$\Rightarrow sk(u) \oplus sk(v) = 0$

→ so let $sk(S) = \bigoplus_{v \in S} sk(S)$ → cancel up if both in S

$= \bigoplus_{\substack{(u,v) \in S \\ v \in S \\ u \notin S}} \langle (u,v) \rangle$

This isn't useful since XORed values doesn't give it to us

Note: The amount of edges leaving S (i.e $cut(S, V-S)$) has values (rounded to factor of 2)

sample the edges so we only have,

$1, 2, 4, 8, \ldots, n^2$

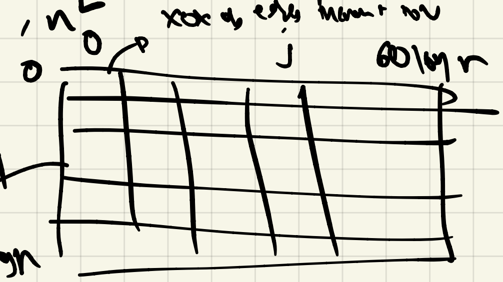
$sk(v) = 2\log n \times 60 \log n$ matrix.

$h_j = \text{edges} \to [0,1]$

→ hash

$sk(v)(i,j) = \bigoplus_{\substack{(u,v) \in E \\ h_j(u,v) \leq 2^{-i}}} \langle u,v \rangle$

xor of edges hash to v



→ xor of 1/4 of edge incident to v

. . . .

Observe) in col j

same hash fun h_i

this last one before zeros $\sim \log_2 \deg(u)$ ish

eventually 0s as your prob too granular

internal cancel

$$\underline{Sk(S)(i,j)} = \bigoplus_{\substack{(u,v) \\ u \in S \\ v \notin S \\ h_j((u,v)) \le 2^i}} \{(u,v)\}$$

size $\log^3 n$

internals will cancel

(the shared random is perfect hash fun! $h_1, \ldots, h_{\log(\ldots)}$)

Lemma) $S \subseteq V$ at least 1 edge in $S, V-S$

$\forall j$ (column) $Pr(\exists i$ so $sk(S)(i,j) = \langle e \rangle$ & e crosses $S, V-S) > .2$

Pf) WLOG crossing edges $\alpha \cdot 2^i$ $\alpha \in [1,2)$

S

$\alpha \cdot 2^i$ cross

Pr (one edges e has prob $h_j(\langle e \rangle) \le 2^{-i}$) $= \alpha 2^i \, 2^{-i} (1-2^{-i})^{\alpha 2^i - 1}$

\# of crossing

prob $h_j \langle e \rangle \le 2^i$

only 1 edge

$1 \le \alpha < 2$

$= \alpha e^{-\alpha} \in [2e^{-2}, e^{-1}]$

$\ge \boxed{0.2}$

When ref sees a string how does it know it legit

Change: $\langle \{u,v\} \rangle = \langle \min\{u,v\}, \max\{u,v\}, g(u,v) \rangle$

g: {edges} $\to$ $10 \log n$ bits

small prob of junk & valid g $\to$ this is a sanity check

$\to$ we can distinguish junk & good edge

So now,

$T = \phi$ → tree

Inv: T is a ~~forest~~ $\subseteq E$ → coll of disj trees

Repeat: $j = 1, \ldots$ 60 log n → columns in $\boxed{SK(S)}$ → can only apply lemma to j once (its useless once we find one exiting edge)

(in parallel) for each $T' \in T$

(sequential) (*) pick an edge $e = (u, v)$ $v \notin V(T')$, $u \in V(T')$

find i s.t. $sk(v(T'))(i, j) = \text{key}$ → if this is one o else, step fails for T'

for each $e = (u,v)$ from (*) if u, v are disc in T

$T = T \cup \{u, v\}$

if * succeeds for k trees reduce # trees by $\boxed{k/2}$

$\boxed{\text{Say}}$ n_j is # non-iso trees after j iter through loop

$n_0 \le n$

random $n_1 = \ldots$

n_2

$\vdots$

$\boxed{n_{jmax}}$

$$E(n_j) \le (n_{i-1}) - \frac{(.2\, n_{i-1})}{2} \qquad .2 \text{ chance of success on} \ldots$$

$$= .9\, n_{i-1}$$

$j_{max} = 60 \log_2 n$

$$E(n_{jmax}) = (0.9)^{jmax}\, n \rightarrow \text{pretty good}$$

$$\le n \cdot n^{-6}$$

$$= \boxed{n^{-5}}$$

fail when $\boxed{n_{jmax} \ne 0}$

$$Pr(n_{jmax} \ge 1) \underset{\text{Markov}}{=} \frac{E(n_{jmax})}{1} = \frac{1}{n^{5}} \rightarrow \text{small}$$