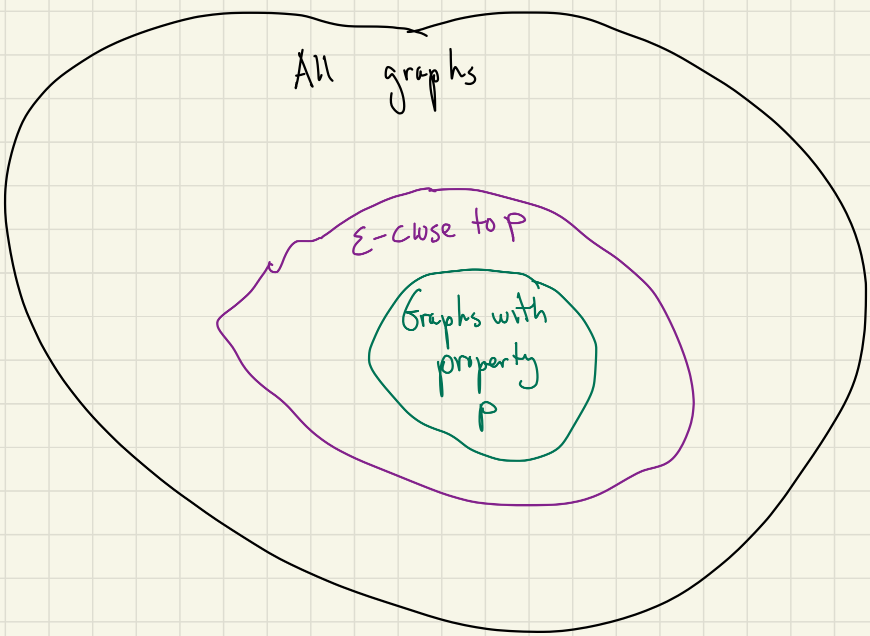
Lecture 6:

Property testing:

is the graph planar?

Property Testing

examples of P:
- planar
- bipartite
- no small cuts
- no triangles
- Connected



Can we distinguish graphs with property P from <u>far</u> from P?

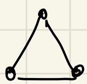
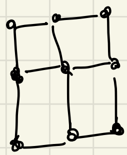
e.g. G is ε-far from planar if must remove $\geq \varepsilon \cdot \Delta \cdot n$ edges to make it planar

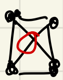
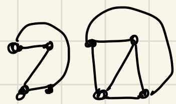
Today's goal:

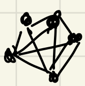
test planarity in time independent of n
(but exponential in ε)

for graphs with max degree Δ

what is a planar graph?
Can be drawn on plane s.t. edges don't intersect

K_3   Yes

$K_{2,2}$  K_4  NO? actually, yes 

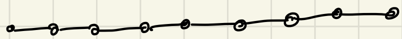
$K_{3,3}$  K_5  NO!

Thm [Kuratowski]

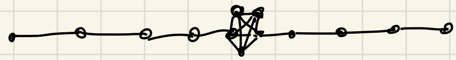
G is planar iff G is $K_{3,3}$ + K_5 minor free

Testing Planarity:

Can't hope to distinguish



from



in sublinear time!

def G is ε-close to planar iff
can remove $\le \varepsilon \cdot \Delta n$ edges to make it

$\left\{ \begin{array}{l} \text{planar} \\ K_{3,3} \,\&\, K_5\text{-free} \end{array} \right.$ $\Leftrightarrow$ equivalent

else G is ε-far

Goal: Given G

- if G planar, PASS
- if G ε-far from planar, FAIL

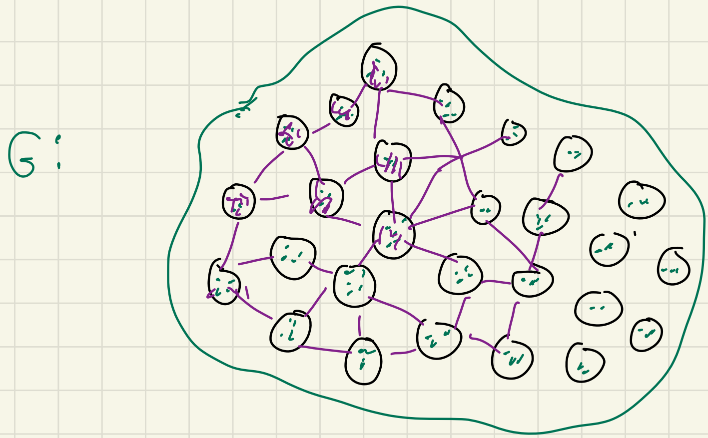
} with prob $\ge 2/3$

$\underbrace{\ge 2/3}_{}$ arbitrary const $\ge 1/2$

Plan for tester: use nice property of Planar graphs

Can always remove $\underbrace{\text{small fraction}}_{\leq \varepsilon}$ of edges

& break up graph into $\underbrace{\text{tiny}}_{\leq \text{const}}$ connected components



def. G is "(ε, k)-hyperfinite" if

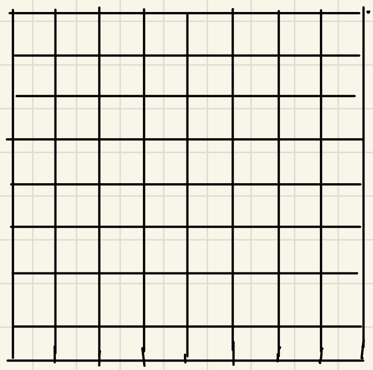
Can remove $\leq \varepsilon n$ edges &

remain with connected components of

size $\leq k$

def. G is "(ε, k)-hyperfinite" if can remove $\le \varepsilon n$ edges & remain with connected components of size $\le k$

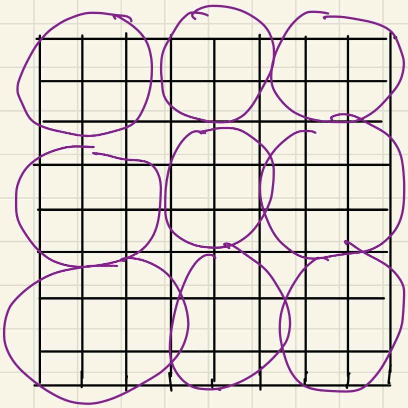
Example: $\overbrace{m \times m}^{n = m^2}$ grid graph



$n = m^2$

example: $m \times m$ Gridgraph

break into $\ell \times \ell$ components



edges crossing component boundaries:

$\leq$ # components $\times$ "surface area"

$\leq \left(\frac{m}{\ell}\right)^2 \times 4 \cdot \ell \leq \frac{4m^2}{\ell}$

pick $\ell = 4/\varepsilon \Rightarrow \left(\varepsilon, \frac{16}{\varepsilon^2}\right)$-h.f.

def. G is "(ε, k)-hyperfinite" if
Can remove $\leq \varepsilon n$ edges &
remain with connected components of
size $\leq k$

Useful Thm

$\forall\ 0 < \varepsilon < 1$, every planar graph G
of deg $\leq \Delta$ is $\left(\varepsilon \cdot \Delta, \frac{c}{\varepsilon^2}\right)$-hyperfinite

remove $\leq \varepsilon \cdot \Delta \cdot n$ edges

Components of size $O\left(\frac{1}{\varepsilon^2}\right)$
no dependence on n

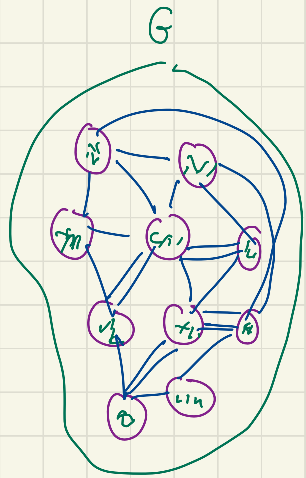
<u>note</u> subgraphs of planar graphs
are also planar, so <u>also</u> hyperfinite

but only remove # edges in proportion to
nodes in subgraph

$\Longrightarrow$ Can recurse + break up further!

Why does hyperfiniteness help in testing?

Plan for testing paradigm:



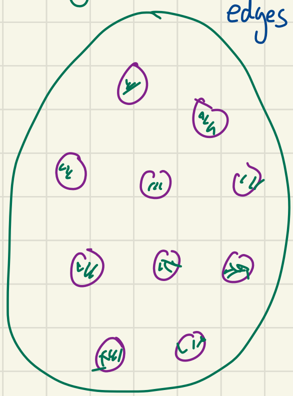
1) Partition graph G into G'

how in sublinear time?

- only const size com. comp. remain
- removed few edges $(\leq \varepsilon \cdot \Delta \cdot n)$
- if can't do this, G is not H-minor free

$\Downarrow$ remove the few blue edges

G'



2) If G' is close to having property so is G

const time

- so test G' by picking random components + seeing if they have the property

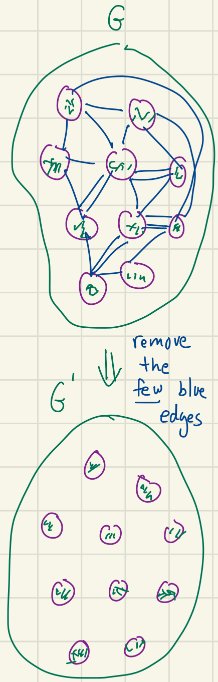
Easy to test since collection of const sized graphs!!

1) Partition graph G into G'

how in sublinear time?
- only const size com. comp. remain
- removed few edges ($\leq \varepsilon \cdot \Delta \cdot n$)
- if can't do this, G is <u>not</u> H-minor free

2) If G' is close to having property so is G

const time
- so test G' by picking random components + seeing if they have the property



Easy to test since collection of const sized graphs!!

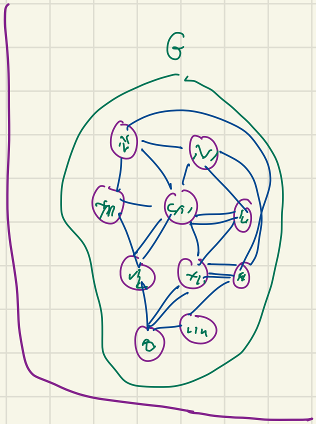
How to determine G'?

- need "local" (sublinear) way to figure out if edge is "blue" (between components) or "green" (inside component)

- will do even better!

  give oracle that tells you "name" of component for each node

Partition Oracle:



input: node v

output: name of v's partition $P[v]$

s.t. $\forall v \in V$ (1) $P[v] \leq k$ (small)

(2) $P[v]$ connected

+ if G is H-minor free

(with prob $\geq 9/10$) (3) $\left|\{(u,v) \in E \mid P(u) \neq P(v)\}\right| \leq \frac{\varepsilon \Delta n}{4}$

few edges cross partitions

Algorithm given Partition Oracle:

I. Does partition oracle give partition that "looks right"?

e.g. few crossing edges

- $\hat{f} \leftarrow$ estimate of # edges (u,v) s.t. $P[u] \neq P[v]$ to additive error $\leq \frac{\varepsilon \Delta n}{8}$ (with prob of failure $\leq \frac{1}{10}$)
- if $\hat{f} > \frac{3}{8}\varepsilon \Delta n$, output "FAIL" + halt

II. Test random partitions

these choose random partitions

- Choose $S = O(1/\varepsilon)$ random nodes
  - if for any $s \in S$, $P[s] \geq k$ or $\underbrace{P[s] \text{ not planar}}_{\text{constant size}}$ reject + halt

constant size $k = O(1/\varepsilon^2)$ so easy to test

III. Accept anything that passed up to this point

Runtime (given oracle):

Part I: $O\left(\frac{1}{\varepsilon^2}\right)$ calls

Part II: $O\Big(\underbrace{\frac{1}{\varepsilon}}_{|S|} \cdot \underbrace{\Delta \cdot k}_{\text{calls for BFS on component of size } \le k \,=\, O\left(\frac{1}{\varepsilon^2}\right)}\Big) = O\left(\frac{\Delta}{\varepsilon^3}\right)$

Algorithm given Partition Oracle:

I. Does partition oracle give partition that "looks right"?

e.g. few crossing edges

- $\hat{f} \leftarrow$ estimate of # edges (u,v) s.t. $P[u] \neq P[v]$ to additive error $\leq \frac{\varepsilon \Delta n}{8}$ (with prob of failure $\leq \frac{1}{10}$)

- if $\hat{f} > \frac{3}{8}\varepsilon \Delta n$, output "FAIL" & halt

II. Test random partitions

these choose random partitions

- Choose $S = O(1/\varepsilon)$ random nodes
  - if for any $s \in S$, $|P[s]| \geq k$ or $\underbrace{P[s] \text{ not planar}}_{\text{constant size}}$ reject & halt

Constant size $k = O(1/\varepsilon^2)$ so easy to test

III. Accept anything that passed up to this point

Behavior (assuming P always "correct"):

- if G is planar:

1) $E[\hat{f}] \leq \frac{\varepsilon \Delta n}{4}$

sampling bounds (Chernoff/Hoeffding) $\Rightarrow$ with prob $\geq 9/10$ $\hat{f} \leq \frac{\varepsilon \Delta n}{4} + \frac{\varepsilon \Delta n}{8} = \frac{3}{8}\varepsilon \Delta n$

$\Longrightarrow$ algorithm doesn't fail stage I with prob $\geq 9/10$

2) $\forall\, s \in V$ $P[s]$ is planar

$\Longrightarrow$ algorithm never fails stage II

$\Longrightarrow$ pass

Algorithm given Partition Oracle:

I. Does partition oracle give partition that "looks right"? e.g. few crossing edges

- $\hat{f} \leftarrow$ estimate of # edges (u,v) s.t. $P[u] \neq P[v]$ to additive error $\leq \frac{\varepsilon \Delta n}{8}$ (with prob of failure $\leq \frac{1}{10}$)
- if $\hat{f} > \frac{3}{8} \varepsilon \Delta n$, output "FAIL" & halt

II. Test random partitions

these choose random partitions

- Choose $S = O(1/\varepsilon)$ random nodes
  - if for any $s \in S$, $|P[s]| \geq k$ or $\underbrace{P[s] \text{ not planar}}_{\text{constant size } k = O(1/\varepsilon^2) \text{ so easy to test}}$ reject & halt

III. Accept anything that passed up to this point

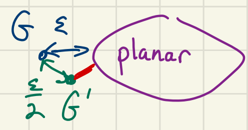
Behavior (assuming P always "correct"):

- if G is ε-far from planar: let $C = |\{(u,v) \in E \mid P[u] \neq P[v]\}|$

Case 1 $C > \frac{\varepsilon \Delta n}{2}$

sampling bnds $\Rightarrow$ $\hat{f} = \frac{\varepsilon \Delta n}{2} - \frac{\varepsilon \Delta n}{8} = \frac{3}{8} \varepsilon \Delta n$

$\Rightarrow$ output "fail" with prob $\geq 9/10$



Case 2 $C < \frac{\varepsilon \Delta n}{2}$

$G' \leftarrow G$ with edges in C removed

since G is ε-far from planar & G' is $\frac{\varepsilon}{2}$-close to G, G' must be $\frac{\varepsilon}{2}$-far from planar if G' is $\frac{\varepsilon}{2}$-far from planar,

must remove $\geq \frac{\varepsilon \Delta n}{2}$ edges

which touch $\geq \frac{\varepsilon n}{2}$ nodes

so with prob $\geq \frac{\varepsilon n}{2}$, pick node in

component which is not planar $\blacksquare$

But how do we implement

P?

Plan for designing Partition Oracle:

1) Define <u>Global</u> partitioning strategy (not sublinear time)

2) Figure out how to implement it <u>locally</u>
(only find partition of a given node, not whole solution)

Useful Concept: "Isolated" neighborhoods

def. G is "(ε, k)-hyperfinite" if can remove $\leq \varepsilon n$ edges & remain with connected components of size $\leq k$

def S is (δ, k)-isolated nbrhd of node v

if
1) $v \in S$
2) S connected
3) $|S| \leq k$
4) # edges connecting S & $\bar{S}$ is $\leq \delta \cdot |S|$

Note: In hyperfinite graphs, most nodes have (δ, k)-isolated nbrhds

obvious?

- G hyperfinite $\Rightarrow \exists$ partitioning BUT will we run into trouble if there is >1 partition and we pick the wrong one?

  NO!

- condition (4) bounds max # cut edges per component, whereas hyperfinite bounds (weighted average

Global Partitioning Algorithm

⇐ a "mental thought process"

def S is (δ, k)-isolated nbrhd of node v

if 1) $v \in S$
2) S connected
3) $|S| \leq k$
4) # edges connecting S + $\bar{S}$ is $\leq \delta \cdot |S|$

- Let $r_1 \cdots r_n$ be nodes in random order
- $P \leftarrow \emptyset$
- For $i = 1$ to n do

  if r_i still in graph then

  use $\delta = \frac{\varepsilon \Delta}{4}$
  $k = O(1/\varepsilon^2)$

  if $\exists$ (δ, k)-isolated nbhd of r_i in remaining graph
  then $S \leftarrow$ this nbhd
  else $S \leftarrow \{r_i\}$

  S is just one node in this case, hopefully doesn't happen often!

  $P \leftarrow P \cup \{S\}$
  remove S + adjacent edges from graph

Does this give partition with few crossing edges?

- if $S \leftarrow (\delta, k)$ isolated nbhd, contributes $\leq \delta |S|$ edges
  overall $\leq \delta n$
- else S is one node: hopefully not often!

Lemma if G' subgraph of hyperfinite graph G s.t. G' has $\geq \delta n$ nodes then $\leq \frac{\varepsilon}{30}$ fraction of nodes in G' don't have (δ, k)-isolated nbhds for $\delta = \varepsilon/30$, $k = \Theta(\varepsilon^3)$

Pf idea: G planar $\Rightarrow$ G' planar $\Rightarrow$ G' hyperfinite

$\exists$ partition s.t. most nodes in G' are in (k, δ)-isolated nbhd

\+

r_i randomly chosen node in G'

$\Downarrow$

whp r_i in (k, δ)-isolated nbhd.

$\Longrightarrow$ not too many "singletons"

∎

Local Simulation of Partitioning Oracle:

- input v
- output $P[v]$
- assume access to random fctn r_v
  $r: V \to [n]$
  random ranking of nodes

- recursively compute $P[w]$ $\forall w$ s.t.
  - w dist $\le k$ from v
  - $r_w \le r_v$ "smaller rank"
- if $\exists$ w s.t. $v \in P[w]$
  then $P[v] = P[w]$

else, look for (k,δ)-isolated nbhd of v
(ignoring any nodes which are already in a partition $P[u]$ for any smaller ranked u)

if find one, $P[v] \leftarrow$ this nbhd

else $P[v] \leftarrow \{v\}$

Query Complexity: using query tree analysis as in last lecture $\Rightarrow 2^{d^{O(k)}}$ for $k = \Theta(1/\varepsilon^3)$
(how many paths of length k survive)

but can do much better:

$d^{O(\log^2(1/\varepsilon))}$ possible