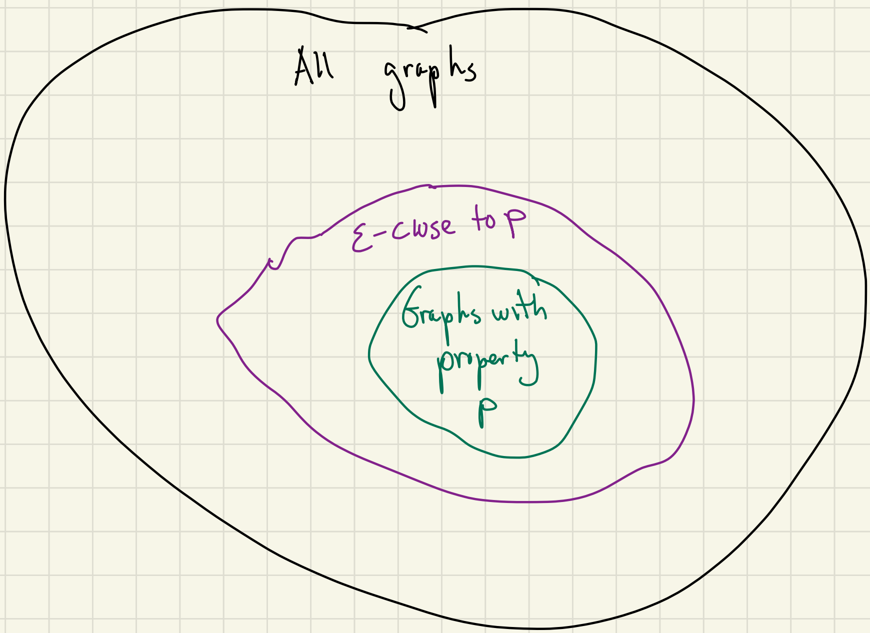
Lecture ~~6~~ 7:

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?

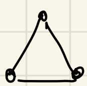
e.g. G is ε-far from planar if must remove $\geq \varepsilon \cdot \Delta \cdot n$ edges to make it planar

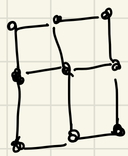
Today's goal:

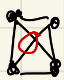
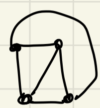
test planarity in time independent of n
(but exponential in ε)

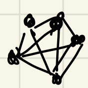
for graphs with max degree Δ

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  

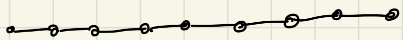
$K_{3,3}$  K_5  NO!

Thm [Kuratowski]

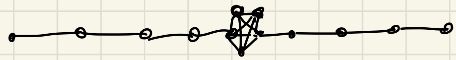
G is planar iff G is $K_{3,3}$ + K_5 minor free

Testing Planarity:

Can't hope to distinguish



from



in sublinear time!

<u>def</u> 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

<u>Goal:</u> Given G

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

$\Big\}$ with prob $\ge 2/3$

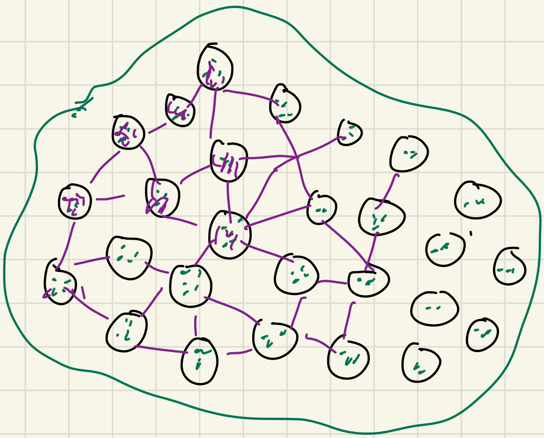
arbitrary const $\ge 1/2$

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

G:



def. G is "(ε, k)-hyperfinite" if

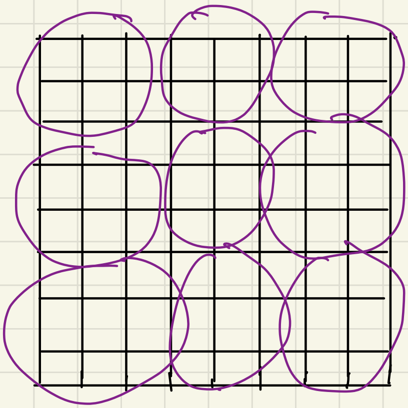
can remove $\leq \varepsilon n$ edges &

remain with connected components of

size $\leq k$

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

Example: $\underbrace{m \times m}_{}$ grid graph $\quad n = m^2$



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 $\left(\underbrace{\varepsilon \cdot \Delta}_{\text{remove } \leq \varepsilon \cdot \Delta \cdot n \text{ edges}}, \underbrace{\frac{c}{\varepsilon^2}}_{\text{components of size } O(\frac{1}{\varepsilon^2}) \text{ no dependence on } n}\right)$-hyperfinite

note: subgraphs of planar graphs are also planar, so also 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'

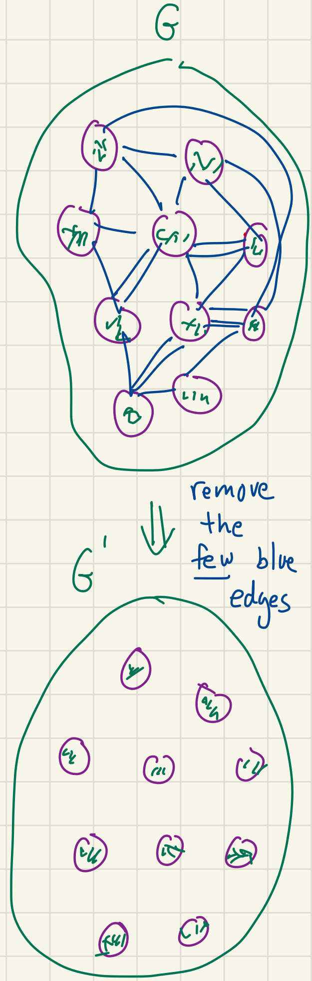
how in sublinear time?

- only const size com. comp. remain
- removed few edges $\left(\le \frac{\varepsilon}{2} \cdot \Delta \cdot n\right)$
- if can't do this, G is <u>not</u> planar.

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!!

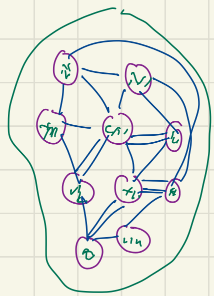
1) Partition graph G into G'

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 not planar

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

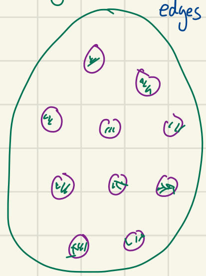
const time
- so test G' by picking random components + seeing if they have the property

G



remove the few blue edges

G'



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)

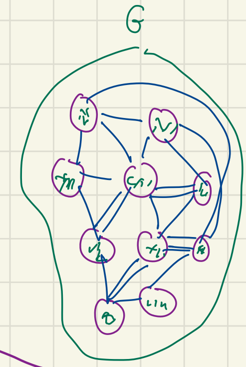
- will do even better!

  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 planar

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

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 $\underbrace{P[s] \text{ not planar}}$
    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: $O\left(\underbrace{\frac{1}{\varepsilon}}_{|S|} \cdot \underbrace{\Delta \cdot k}_{\text{calls for BFS on component of size } \leq k \,=\, O\left(\frac{1}{\varepsilon^2}\right)}\right) = O\left(\frac{\Delta}{\varepsilon^3}\right)$

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

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

III. Accept anything that passed up to this point

Behavior (assuming P always "correct"):

- if G is planar:

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

with prob $\geq 9/10$

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

$\Longrightarrow$ 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 $P[s]$ not planar reject & halt

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

III. Accept anything that passed up to this point

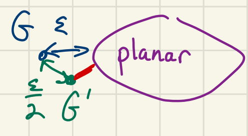
Behavior (assuming P always "correct"):

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

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

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 $\geq 9/10$



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

$G' \leftarrow G$ 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?

Partition Oracle:

input: node v

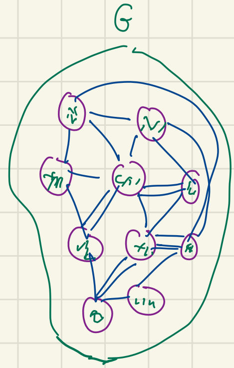
output: name of v's partition $P[v]$

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

(2) $P[v]$ connected

+ if G is planar

(with prob $\ge 9/10$) (3) $\left| \{ (u,v) \in E \mid P(u) \ne P(v) \} \right| \le \frac{\varepsilon \Delta n}{4}$ } few edges cross partitions



Plan for designing Partition Oracle:

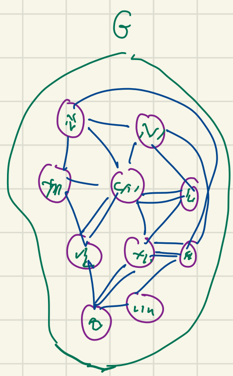
1) Define <u>Global</u> partitioning strategy (not sublinear time)

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

Partition Oracle:

input: node v

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



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

(2) $P[v]$ connected

+ if G is H-minor free

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

few edges cross partitions

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 $\leq \delta \cdot |S|$

edges attached to S that cross cut

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

obvious?

- G hyperfinite $\Rightarrow$ $\exists$ partitioning but some partitions might have lots of edges
  but on average, they don't have many
  so most don't have many
- is it an issue that there could be many partitions?

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) $|S| \le k$
4) # edges connecting S & $\bar{S}$ is $\le \delta \cdot |S|$

- 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

  if $\exists$ (δ, k)-isolated nbhd of r_i in remaining graph

  then $S \leftarrow$ this nbhd

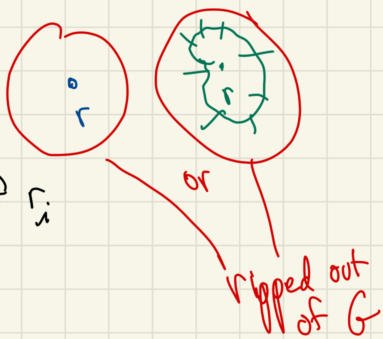
  else $S \leftarrow \{r_i\}$ "singleton"

  $P \leftarrow P \cup \{S\}$

  remove S & adjacent edges from graph

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

S is just one node



Does this give partition with few crossing edges?

- if $S \leftarrow (\delta, k)$ nbhd, contributes $< \delta |S|$ edges, overall $\le \delta n$ edges from these additions
- else S is one node, no bound
  hopefully doesn't happen often

Lemma if G' subgraph of planar graph G

then $\le \frac{\varepsilon}{30}$ fraction of nodes in G' become singletons

Pf idea: (weaker theorem + parameters a bit off)

- in loop i, call remaining graph $G^{(i)}$.

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

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

- $G^{(i)}$ is planar $\Rightarrow$ hyperfinite $\Rightarrow \exists\ (\frac{1}{60}, k)$-partition of $G^{(i)}$ into $G^{(i)'}$ (new partition at each loop)

low wt nodes are not singletons
- assign each node in partition S the weight $\frac{\#\text{edges connecting } S + \bar{S}}{|S|}$
- total weight of nodes in S = # edges connecting $S + \bar{S}$
- total weight of nodes in $G^{(i)}$ = 2 $\cdot$ # removed edges in partition of $G^{(i)}$ into $G^{(i)'}$ $\le 2 \cdot \frac{1}{60} \cdot n$
- average weight of node $\le \frac{2}{60} = \frac{1}{30}$
- # nodes with weight $\ge \frac{1}{30} \cdot \frac{30}{\varepsilon} = \frac{1}{\varepsilon}$ is at most $\frac{\varepsilon n}{30}$ by Markov's $\neq$

r_i is either not in $G^{(i)}$ + therefore not in $G^{(i)'}$

or r_i is uniform node in $G^{(i)}$ + $G^{(i)'}$ so hits one of $\underbrace{\text{"low weight"}}_{\text{these are not singletons}}$ nodes with prob $1 - \frac{\varepsilon}{30}$

Over all rounds/loops, expect that $\le \frac{\varepsilon}{30}$ fraction become 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 $\leq k$ from v
  - $r_w \leq r_v$
- if $\exists$ w s.t. $v \in P[w]$
  then $P[v] = P[w]$

else, look for (k, δ)-isolated nbhd of v
(ignore any nodes removed by lower ranked w)

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

Query Complexity:

use same analysis as for simulating greedy
(as in 2 lectures ago)

$2^{d^{O(k)}}$ for $k = O(1/\varepsilon^3)$

can do much better! $d^{O(\log^2(1/\varepsilon))}$ possible