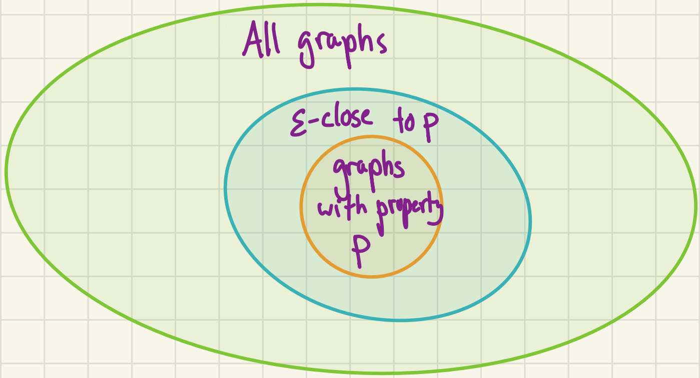
Lecture 6:

- Property Testing
- Testing Planarity
- Partition oracles

Property Testing



P is a subset of graphs

Can we distinguish graphs in P from graphs that are not in P? not even ε-close?

Goal if G has property P, pass
if G ε-far from P, fail

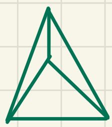
(if G is ε-close, can either pass or fail)

For today: <u>def</u> deg $\leq d$ graph G is ε-close to P if can remove some $\leq \varepsilon d n$ edges to turn G into $G' \in P$

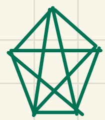
Planarity:

def a Planar graph can be drawn in plane s.t. edges intersect only at endpts.

e.g.



planar



not planar

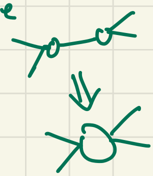
Cool Thm [Kuratowski]

G is planar iff does not contain K_5 or $K_{3,3}$ as minor

K_5 ← complete graph on 5 nodes

$K_{3,3}$ ← complete bipartite graph with 3 nodes on side

minor ← subgraph repeatedly contract edges into nodes



Hyperfiniteness

def G is (ε, k)-hyperfinite if
can remove $\leq \varepsilon n$ edges
+ remain with all connected components
of size $\leq k$. ← k can be fctn of ε

remove few edges + break up graph into
tiny pieces.

Important Thm $\forall\, \varepsilon, d,$ $\exists$ constant c s.t.
every planar graph of max degree d
is $(\varepsilon d,\ c/\varepsilon^2)$-hyperfinite

Note subgraphs of deg $\leq d$ planar graphs
are also deg $\leq d$ planar graphs
$\Rightarrow$ also hyperfinite

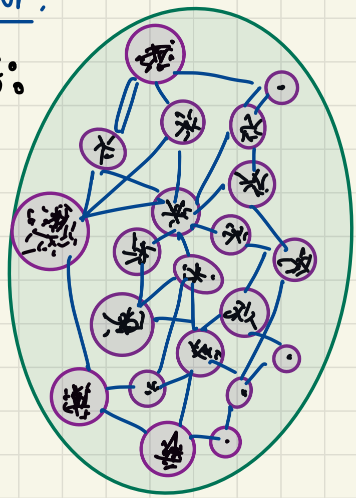
Why is hyperfiniteness useful?

how in sublinear time?

Partition G with parameter $\frac{\varepsilon}{2} \cdot d$

$G' \leftarrow G$ minus edges between partitions

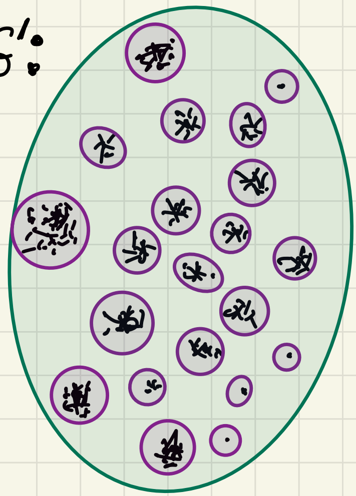
G:



nice properties of G':

- G planar $\Rightarrow G'$ planar

G':



- G' is very similar to G:

  differ by $\leq \frac{\varepsilon}{2} \cdot d$ edges

  $\Rightarrow$ if G is ε-far from planar then G' is $\frac{\varepsilon}{2}$-far from planar

- All connected components in G' are size $O(1/\varepsilon^2)$

Can test for planarity in time indep of n

If there is no such partition of G, then G is not planar!! but, can we find it in sublinear time?

Partition Oracle

(use slightly different parameters from previous)

input v ← node

output $P[v]$ ← name of v's partition

s.t. $\forall\, v \in V$ (1) $|P[v]| \le k$ (2) $P[v]$ connected } partitions small & connected

& if G planar then

(with prob $\ge \frac{9}{10}$) $\underbrace{|\{(u,v) \in E \mid P(u) \neq P(v)\}|}_{\text{edges crossing partitions}} \le \underbrace{\frac{\varepsilon d n}{4}}_{\text{small fraction}}$

Note for planar graphs there is at least one P, but there could be many possible partitions P. The oracle doesn't have to decide "in advance" which partition to use, but must stay consistent.

Algorithm given partition oracle P

assume it always works for planar G

I. Does P 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 within additive error $\leq \frac{\varepsilon d n}{8}$ with prob of failure $(\delta) \leq \frac{1}{10}$
- if $\hat{f} \geq \frac{3}{8} \varepsilon d n$, output "not planar" & halt

II. Test random partitions for planarity

- Choose $S = O(\frac{1}{\varepsilon})$ random nodes
  - $\forall s \in S$ if $P[s] \geq k$ $\equiv 4\varepsilon^2$ or $P[s]$ not planar } size $\leq k$ so easy to test
    output "not planar" & halt
- Output "planar"

Runtime: part I: $O(\frac{1}{\varepsilon^2})$ calls to oracle

part II: $O(d/\varepsilon^2)$ calls to determine $P[S]$ via BFS

$O(d/\varepsilon^3)$ total calls

Analysis: (assume oracle works for planar G)

def $C \equiv \{(u,v) \in E \mid P(u) \neq P(v)\}$ "edges that cross between partitions"

I. if G is planar, $\exists$ good partition & by assumption that oracle works, # edges crossing partition $\leq \frac{\varepsilon d n}{4}$

so $E[\hat{f}] \leq \frac{\varepsilon d n}{4}$

$\Longrightarrow$ (by Chernoff/Hoeffding) $\hat{f} \leq \frac{\varepsilon d n}{4} + \frac{\varepsilon d n}{8} = \frac{3 \varepsilon d n}{8}$ with prob $\geq 9/10$

$\Longrightarrow$ algorithm continues to part II with prob $\geq 9/10$

Also, $\forall s \in V$, $P[S]$ is Planar

$\Longrightarrow$ output "Planar" with prob $\geq 9/10$

II. If G is ε- far from planar:

<u>Case 1</u> Partition P doesn't satisfy $|C| < \frac{\varepsilon d n}{2}$:

sampling bounds $\Rightarrow$ $\hat{f} > \frac{\varepsilon d n}{2} - \frac{\varepsilon d n}{8} = \frac{3}{8} \varepsilon d n$

$\Rightarrow$ output "not planar" with prob $\geq 9/10$

<u>Case 2</u> P satisfies $|C| < \frac{\varepsilon d n}{2}$:

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

<u>note</u>: G' is $\frac{\varepsilon}{2}$-close to G

so G is ε-far from planar,

$\Rightarrow$ G' is $\frac{\varepsilon}{2}$-far from planar

issue: we are picking random nodes in part II, not random edges.

<u>but</u> graph is degree $\leq d$

since G' $\frac{\varepsilon}{2}$-far from planar, must change $\geq \frac{\varepsilon}{2} \cdot dn$ edges, which touch $\geq \frac{\varepsilon}{2} \cdot n$ nodes so with prob $\geq \frac{\varepsilon}{2} \cdot n$, pick node in component which is not H-minor free. $\blacksquare$

Remaining issue:

How do we implement $\mathcal{P}$?

Plan:

1) Define <u>global</u> partitioning strategy
2) Figure out how to <u>locally</u> implement (only find partition of given node, not whole solution).

Useful concept: Isolated Neighborhoods

<u>def</u> S is (δ, k)-isolated neighborhood of node v if:

(1) $v \in S$

(2) S connected

(3) $|S| \leq k$

(4) # edges connecting S & $\bar{S} \leq \delta |S|$

↑ $\delta < 1$ but degree bound only gives $d \cdot |S|$

<u>Observe</u> in planar graphs, most nodes have (δ, k)-isolated nbhds in any good partition

$\varepsilon \cdot d \to \delta$, $k \leftarrow c/\varepsilon^2$

obvious? yes, on <u>average</u> but

Planar G is hyperfinite $\Rightarrow \exists$ partition with few <u>total</u> crossing edges?

but maybe some partitions have lots of edges coming out? <u>still</u> most have close to average. (Markov's ≠) * think at home

Will need observation to be true
in context of evolving "step-by-step"
partition.

Luckily, graph stays planar | hyperfinite as
evolve (remove nodes)

Global Partitioning Algorithm ← mental thought process

Let $\pi_1 \dots \pi_n$ be nodes in random order

$P \leftarrow \emptyset$

For $i = 1..n$ do

if π_i still in graph then

BFS to distance k of π_i

if $\exists\,(\delta,k)$-isolated nbhd of π_i in remaining graph

then $S \leftarrow$ this nbhd

else $S \leftarrow \{\pi_i\}$ ⇐ S is just one node. hopefully doesn't happen often!

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

Remove S & adjacent edges from G

Behavior: few crossing edges?

- S s.t. S is (δ,k)-isolated contribute $\leq \delta|S|$ edges

  $\Rightarrow$ overall $\sum \delta |S_i| \leq \delta \cdot n$

- S s.t. $S = \{\pi_i\}$ (one node)

  need to show not too many!!

$|V| = n$

Lemma if $G' = (V', E')$ subgraph of planar $G = (V, E)$

s.t. $|V'| \geq \delta n$

then $\leq \frac{\varepsilon}{30}$ fraction of V' don't have

(δ, k)- isolated nbhds for $\delta = \varepsilon/30$

$k = \Theta(1/\varepsilon^3)$ } need to use stronger settings in "important theorem"

Pf idea G planar

$\Downarrow$

G' planar

$\Downarrow$

G' hyperfinite

$\Downarrow$

$\exists$ partition s.t. most nodes in (k, δ)- isolated nbhd

Markov's $\neq$

$+$

Π_i randomly chosen

$\Downarrow$

whp Π_i in (k, δ)- isolated nbhd

∎

So, not too many "singletons"

Local Simulation of Partition Oracle

- input v
- assume access to random fctn $\pi(v)$ s.t. $\pi: V \to [n]$
- Output $P[v]$

Algorithm on input v

I. recursively compute $P[w]$ $\forall$ w s.t.

$\pi(w) < \pi(v)$

& dist w from $v \leq 2k$

$\}$ $d^{O(k)}$ of these

II. if $\exists$ w s.t. $v \in P[w]$ ← $P[v]$ already decided by earlier w

(A) then $P[v] = P[w]$

else look for (k, δ)- isolated nbhd of v

(B) (ignoring nodes in $P[w]$ for smaller w's)

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

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

Implementing algorithm:

step I:

$d^{O(k)}$ recursive computations on lower ranked nodes. Analysis similar to last lecture $\Rightarrow$ $2^{d^{O(k)}} + k \approx O(1/\varepsilon^3)$

step II: (A) computed in step I

(B) figure out remaining nodes w/in dist K from step I.

Brute force on graph of size $\leq d^{O(k)}$

Can do much better:

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

(may have been improved)