6.5240 Sublinear Time Algorithms	October 8, 2026
Lecture 9	
Lecturer: Ronitt Rubinfeld	*Scribe: Aaron Martin*

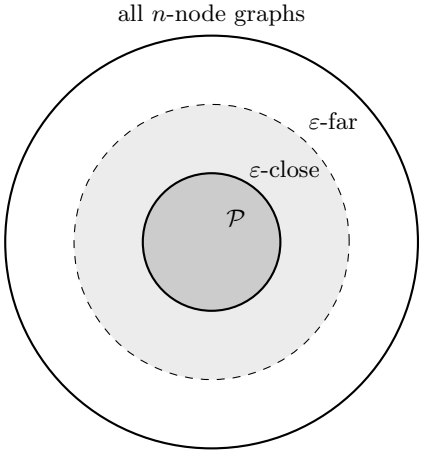
Overview. We provide a planarity testing algorithm for graphs of maximum degree $\leq \Delta$, using the fact that degree-bounded planar graphs are $(\varepsilon\Delta, c/\varepsilon^2)$-hyperfinite. Given a partition oracle that provides local access to a partitioning of a graph, we give a sublinear algorithm that uses $O(\Delta/\varepsilon^3)$ oracle queries.

Topics.

- Property testing Model
- Planar graphs
- Hyperfinite graphs
- Partition oracles
- Planarity testing algorithm
- Isolated neighborhoods
- Additionally, proof of hyperfiniteness of degree-bounded planar graphs

1 Property Testing

Property testing relaxes a binary decision problem into an approximation problem. Rather than decide whether a graph G has property $\mathcal{P}$, a property tester distinguishes graphs that have $\mathcal{P}$ from graphs that are far from having it.



Let G be an n-node graph. A property tester has the following goals.

1. If $G \in \mathcal{P}$, output Pass with $\mathbb{P} \geq \frac{2}{3}$
2. If G is ε-far from $\mathcal{P}$, output Fail with $\mathbb{P} \geq \frac{2}{3}$
3. If G is ε-close to $\mathcal{P}$ (but not in $\mathcal{P}$), output anything

We will analyze graphs with maximum degree $\leq \Delta$ and define closeness as follows.

Definition 1 (ε-close)**.** A degree $\leq \Delta$ graph G is *ε-close* to having property $\mathcal{P}$ if $\leq \varepsilon\Delta n$ edges can be removed to turn G into some $G' \in \mathcal{P}$.

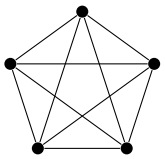
- $\leq \varepsilon\Delta n$ as a bound for "ε-close"ness is good for sparse graphs where edges are being mutated and $|E| \in O(|V|)$, dense graphs would need bounds $\leq \varepsilon n^2$

2 Planar Graphs

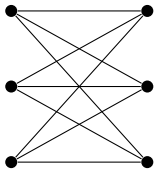
Definition 2 (Planar)**.** A graph is *planar* if it can be drawn in the plane such that edges only intersect at their endpoints.

Definition 3 (Minor)**.** A *minor H* of graph G is a graph obtained from G by a sequence of removing nodes, removing edges, or contracting edges.

Definition 4 (K_5)**.** K_5 is the complete graph on five nodes.



Definition 5 ($K_{3,3}$)**.** $K_{3,3}$ is the complete bipartite graph with three nodes on each side: every node on one side is joined to every node on the other side, and no two nodes on the same side are joined.



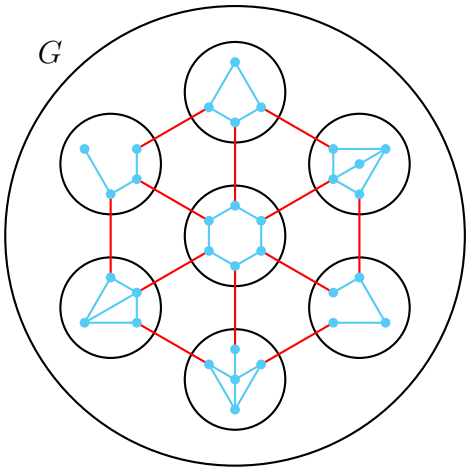
2.1 Properties of Planar Graphs

- **Subgraphs:** Removing nodes or edges preserves planarity.
- **Minors:** Contracting edges (merging two endpoints sharing an edge into one) preserves planarity.
- **Sparse:** Has at most $3n - 6$ edges for $n \geq 3$.
- **Degree:** Average degree less than 6.
- G is planar if and only if G is K_5-minor-free and $K_{3,3}$-minor-free.

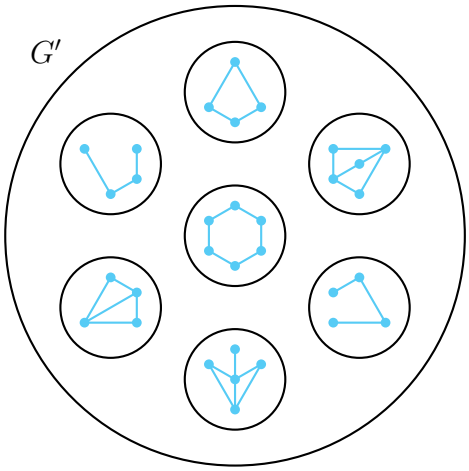
2.2 Hyperfinite Graphs

Definition 6 (Hyperfinite)**.** A graph G is (ε, k)*-hyperfinite* if at most εn edges can be removed and the resulting subgraph G' contains connected components of size $\leq k$.

Theorem 7 (Hyperfiniteness of degree-bounded planar graphs)**.** *For all $0 < \varepsilon \leq 1$ and $\Delta > 1$, there exists a constant c such that every planar graph of maximum degree $\leq \Delta$ is $(\varepsilon\Delta, \frac{c}{\varepsilon^2})$-hyperfinite.*



- Number of edges crossing components is $\leq \varepsilon \Delta n$
- Size of each component is $\leq \frac{c}{\varepsilon^2}$
- A hyperfinite graph does not imply its subgraphs are hyperfinite, unless the graph is planar



- $G = (V, E)$ becomes $G' = (V, E \setminus \{\text{edges crossing components}\})$
- distance of G and G' is $\leq \varepsilon \Delta n$ edges
- if G is planar, then G' is planar
- if G is far from planar, G' also has to be pretty far from planar

3 Algorithm for Testing Planarity

Idea. Check subgraphs of G' for planarity to estimate whether G is planar. Since we cannot afford to compute a hyperfinite partition of all of G, we use a partition oracle that gives local access to a hyperfinite partition: it names the piece a node belongs to without calculating the whole partition.

3.1 Partition Oracle

A partition oracle P for G with parameter ε answers the following queries.

- Input: a node v
- Output: $P[v]$, the name of v's partition, such that for all $v \in V$
  1. $|P[v]| \leq k = \frac{c}{\varepsilon^2}$, i.e. the partition containing v has at most k nodes
  2. $P[v]$ is connected
- If G is planar, with $\mathbb{P} \geq \frac{9}{10}$, $|\{(u, v) \in E : P[u] \neq P[v]\}| \leq \frac{\varepsilon \Delta n}{4}$ i.e. the number of red edges is bounded

Such a partition exists because planar graphs are hyperfinite. If G is not planar, G is ε-far or ε-close and the algorithm should output NOT PLANAR for ε-far graphs with high probability.

3.2 Algorithm

Given partition oracle P, assume that if G is planar, the partition oracle returns a valid partition (red edges $\leq \frac{\varepsilon\Delta n}{4}$) with $\mathbb{P} = 1$. Denote the set of red edges R.

$$R = \{(u, v) \in E : P[u] \neq P[v]\}$$

The algorithm runs in two steps **I** and **II** each testing for a specific quality of G to determine planarity.

I. Does P's partition look right? A large $|R|$ occurs when

1. G is not planar
2. the partition oracle fails, which happens with low probability. Under our assumption that the partition oracle returns a valid partition with $\mathbb{P} = 1$, this cannot occur.

A large $|R|$ indicates P's partition does not look right and thus G is not planar.

1. Sample m nodes $v_1, v_2, \ldots, v_m$ uniformly at random.
2. For each v_i, query $P[v_i]$, and for each neighbor $w \in N(v_i)$ of v_i query $P[w]$. Let X_i be the number of neighbors w that are not in the same component as v_i.

$$X_i = |\{w \in N(v_i) : P[w] \neq P[v_i]\}| \in [0, \Delta]$$

3. Let $\hat{f}$ be an estimator of $|R|$

$$\hat{f} = \frac{n}{2m} \sum_{i=1}^{m} X_i$$

4. If $\hat{f} \geq \frac{3\varepsilon\Delta n}{8}$, output NOT PLANAR and halt.

II. Test random components for planarity. **I** passes graphs that can be partitioned accurately by the oracle but says nothing about planarity of the subgraphs.

1. Sample l nodes $v_1, v_2, \ldots, v_l$ uniformly at random.
2. For each v_i, get its corresponding subgraph by running a breadth-first search (BFS) from v_i that only follows edges (u, w) with $P[w] = P[v_i]$. Run a deterministic planarity testing algorithm on the subgraph of at most k nodes and output NOT PLANAR if the subgraph fails.
3. Output PLANAR if all sampled subgraphs are planar.

3.3 Query Complexity

Each query is one call to the partition oracle.

I. Chernoff bounds give that $m = O(1/\varepsilon^2)$ samples are enough to estimate $|R|$ to within $\pm\frac{\varepsilon\Delta n}{8}$. Each sample costs $1 + \Delta$ calls, one for $P[v_i]$ and one for each neighbor, so **I** makes $O(\Delta/\varepsilon^2)$ calls to the oracle.

II. There are $l = O(1/\varepsilon)$ BFS runs, each visiting at most k nodes and querying at most Δ neighbors per node, so **II** makes $O(k \cdot \Delta/\varepsilon) = O(\Delta/\varepsilon^3)$ calls, where k comes from the size of a component and Δ from the BFS. The planarity test of a component uses no queries and takes $O(k)$ time.

Total. $O(\Delta/\varepsilon^3)$ calls to the oracle.

3.4 Analysis

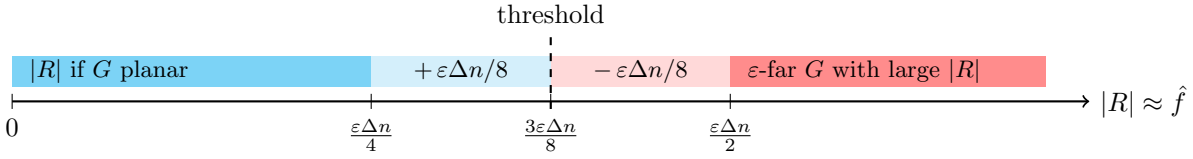
Let G' be G with R removed, so G' keeps only the blue edges and its connected components are exactly the partitions.

- $\text{dist}(G, G') = |R|$, where dist is the number of edges on which two graphs differ, since exactly the red edges were removed.
- If G is planar, then G' is planar, since removing edges preserves planarity.
- If G is far from planar, then G' also has to be pretty far from planar. Any removed set of edges that makes G' planar, together with R, makes G planar. So fixing G' must cost more than $\varepsilon\Delta n - |R|$ edges.

Where $\frac{3\varepsilon\Delta n}{8}$ comes from. **I** has to tell apart two situations using an estimate $\hat{f}$ of $|R|$ that can be off by up to $\frac{\varepsilon\Delta n}{8}$:

- *G planar:* the oracle guarantees $|R| \le \frac{\varepsilon\Delta n}{4}$, so after sampling error $\hat{f} < \frac{\varepsilon\Delta n}{4} + \frac{\varepsilon\Delta n}{8} = \frac{3\varepsilon\Delta n}{8}$.
- *G ε-far with many red edges:* $|R| \ge \frac{\varepsilon\Delta n}{2}$, so after sampling error $\hat{f} > \frac{\varepsilon\Delta n}{2} - \frac{\varepsilon\Delta n}{8} = \frac{3\varepsilon\Delta n}{8}$.

I should pass planar graphs with high probability. It should also catch all ε-far graphs that have many red edges because **II** catches ε-far graphs with few red edges $|R| < \frac{\varepsilon\Delta n}{2}$.



I. G planar. There exists a good partition, and by assumption the oracle P finds it, so $|R| \leq \frac{\varepsilon\Delta n}{4}$. If the estimate $\hat{f}$ is good, which happens with $\mathbb{P} \geq \frac{9}{10}$,

$$\hat{f} < \underbrace{\frac{\varepsilon\Delta n}{4}}_{\text{oracle bound}} + \underbrace{\frac{\varepsilon\Delta n}{8}}_{\text{additive error}} = \frac{3\varepsilon\Delta n}{8}.$$

The algorithm outputs PLANAR. A planar graph is never rejected by **II**, so the only error on planar graphs comes from a bad estimate in **I**.

II. G ε-far from planar. The oracle makes no promise about $|R|$, so there are two ways the non-planarity can show up.

Case 1: Large $|R| \geq \frac{\varepsilon\Delta n}{2}$ (found in I). P does not give a good partition. With $\mathbb{P} \geq \frac{9}{10}$ the sampling is good, and

$$\hat{f} > |R| - \frac{\varepsilon\Delta n}{8} \geq \frac{\varepsilon\Delta n}{2} - \frac{\varepsilon\Delta n}{8} = \frac{3\varepsilon\Delta n}{8}.$$

Even if the sample undercounts, the partition is so bad that **I** outputs NOT PLANAR.

Case 2: $|R| < \frac{\varepsilon\Delta n}{2}$ (found in II). Because fixing G' costs more than $\varepsilon\Delta n - |R|$ edges and $|R| < \frac{\varepsilon\Delta n}{2}$, we show that a random sample of partitions is likely to include a non-planar partition.

- G' is $\frac{\varepsilon}{2}$-far, so making it planar requires removing more than $\frac{\varepsilon\Delta n}{2}$ edges.
- Deleting every edge inside the non-planar components makes G' planar, which means those components contain more than $\frac{\varepsilon\Delta n}{2}$ edges.
- By the degree bound, each node touches at most Δ edges, so these edges touch more than $\frac{\varepsilon\Delta n/2}{\Delta} = \frac{\varepsilon n}{2}$ nodes. Call this set of nodes in non-planar components B.
- The sample of l nodes in **II** will likely contain a node of B with $\mathbb{P} \geq \frac{9}{10}$. The BFS recovers that node's whole component, the planarity test fails, and **II** outputs NOT PLANAR.

Note. Degree-bounded nodes cannot be dense in one area that is not sampled: the non-planarity cannot hide in a few nodes, because each node can only account for Δ of the more than $\frac{\varepsilon\Delta n}{2}$ bad edges.

Conclusion. If we have partition oracles, the algorithm outputs PLANAR on planar graphs with $\mathbb{P} \geq \frac{9}{10}$ and NOT PLANAR on ε-far graphs with $\mathbb{P} \geq \frac{9}{10}$, using $O(\Delta/\varepsilon^3)$ oracle calls.

3.5 Sample Sizes m and l

Claim (sample size m). With $m = \lceil 24/\varepsilon^2 \rceil$,

$$\mathbb{P}\left[\left|\hat{f} - |R|\right| < \frac{\varepsilon \Delta n}{8}\right] \geq \frac{9}{10}.$$

Proof.

1. $\hat{f}$ is unbiased. Each red edge is counted once from each of its endpoints, so $\sum_{v \in V} X_v = 2|R|$. Since each v_i is uniform over V,

$$\mathbb{E}[X_i] = \frac{1}{n} \sum_{v \in V} X_v = \frac{2|R|}{n}, \qquad \mathbb{E}[\hat{f}] = \frac{n}{2m} \sum_{i=1}^{m} \mathbb{E}[X_i] = |R|.$$

2. Normalize to $[0, 1]$ since Chernoff bounds need bounded variables, so let $Y_i = X_i/\Delta \in [0, 1]$, $\bar{Y} = \frac{1}{m} \sum_i Y_i$, and $\mu = \mathbb{E}[Y_i] = \frac{2|R|}{\Delta n}$. Then $\hat{f} = \frac{\Delta n}{2} \bar{Y}$, so the bad event becomes

$$\left|\hat{f} - |R|\right| \geq \frac{\varepsilon \Delta n}{8} \iff \frac{\Delta n}{2} |\bar{Y} - \mu| \geq \frac{\varepsilon \Delta n}{8} \iff |\bar{Y} - \mu| \geq \frac{\varepsilon}{4}.$$

3. Additive Chernoff (Hoeffding) bound. For independent $Y_i \in [0, 1]$, $\mathbb{P}[|\bar{Y} - \mu| \geq t] \leq 2e^{-2mt^2}$. With $t = \frac{\varepsilon}{4}$,

$$\mathbb{P}\left[\left|\hat{f} - |R|\right| \geq \frac{\varepsilon \Delta n}{8}\right] = \mathbb{P}\left[|\bar{Y} - \mu| \geq \frac{\varepsilon}{4}\right] \leq 2e^{-2m(\varepsilon/4)^2} = 2e^{-m\varepsilon^2/8}.$$

4. Solve for m. We want the bad event to have $\mathbb{P} \leq \frac{1}{10}$:

$$2e^{-m\varepsilon^2/8} \leq \frac{1}{10} \iff m \geq \frac{8 \ln 20}{\varepsilon^2} \approx \frac{23.97}{\varepsilon^2},$$

so $m = \lceil 24/\varepsilon^2 \rceil$ suffices, giving $2e^{-m\varepsilon^2/8} \leq 2e^{-3} \approx 0.0996$.

5. Take the complement.

$$\mathbb{P}\left[\left|\hat{f} - |R|\right| < \frac{\varepsilon \Delta n}{8}\right] = 1 - \mathbb{P}\left[\left|\hat{f} - |R|\right| \geq \frac{\varepsilon \Delta n}{8}\right] \geq 1 - 2e^{-3} > 1 - \frac{1}{10} = \frac{9}{10}.$$

□

Claim (sample size l). Let $B \subseteq V$ with $|B| > \frac{\varepsilon n}{2}$, and let S be a set of $l = \lceil 5/\varepsilon \rceil$ nodes sampled uniformly at random. Then

$$\mathbb{P}\left[S \cap B \neq \emptyset\right] \geq \frac{9}{10}.$$

Proof.

1. One sample misses B with $\mathbb{P} = 1 - \frac{|B|}{n} < 1 - \frac{\varepsilon}{2}$.

2. All l independent samples miss B. Using $1 - x \le e^{-x}$,

$$\mathbb{P}[S \cap B = \emptyset] < \left(1 - \frac{\varepsilon}{2}\right)^l \le e^{-\varepsilon l/2}.$$

3. Solve for l. We want the bad event to have $\mathbb{P} \le \frac{1}{10}$:

$$e^{-\varepsilon l/2} \le \frac{1}{10} \iff l \ge \frac{2 \ln 10}{\varepsilon} \approx \frac{4.61}{\varepsilon},$$

so $l = \lceil 5/\varepsilon \rceil$ suffices, giving $e^{-\varepsilon l/2} \le e^{-5/2} \approx 0.082$.

4. Take the complement.

$$\mathbb{P}[S \cap B \neq \emptyset] = 1 - \mathbb{P}[S \cap B = \emptyset] > 1 - e^{-5/2} > 1 - \frac{1}{10} = \frac{9}{10}.$$

□

4 Constructing a Partition Oracle

We briefly discuss how a partition oracle can be constructed.

Definition 8 (Isolated Neighborhood)**.** A set $S \subseteq V$ is a (δ, k)-*isolated neighborhood* of node v if

1. $v \in S$
2. S induces a connected subgraph
3. $|S| \le k$
4. the number of edges between S and $V \setminus S$ is $\le \delta |S|$

The partition oracle will find isolated neighborhoods, and if it cannot associate a node with one, it will output that node as a singleton set.

Note that this is a stronger requirement than the partition oracle bounding the total number of crossing edges between components, because it says every component has a small number of edges coming out of it.

5 Appendix

5.1 Proving Degree-Bounded Planar Graphs are Hyperfinite

The proof uses the Lipton–Tarjan planar separator theorem to repeatedly cut the graph and remove nodes until the remaining components' sizes are bounded.

5.1.1 Lipton–Tarjan Planar Separator Theorem

Lemma 9 (Lipton–Tarjan)**.** *The nodes of every planar graph G with n nodes can be partitioned into three sets A, B, and S such that*

1. *no edge of G joins a node of A to a node of B*
2. *$|A| \leq \frac{2n}{3}$ and $|B| \leq \frac{2n}{3}$*
3. *$|S| \leq 7\sqrt{n}$*

Proof. The proof follows by using two facts about planar graphs.

- **Jordan Curve Theorem.** In a planar graph, every cycle C cuts the plane into two regions, and every node not on C lies in exactly one of them. No edge joins a node in one region to a node in the other, since such an edge would cross C. The nodes in one region form A, the nodes in the other region form B, and the nodes of the cycle form the separator S.
- **Triangulation.** If $n \geq 3$, edges can be added to a planar graph, keeping it planar and simple, until every face is a triangle. Since no edge can be added to a triangulated planar graph, by the maximum edge property of planar graphs, this graph has $3n - 6$ edges.

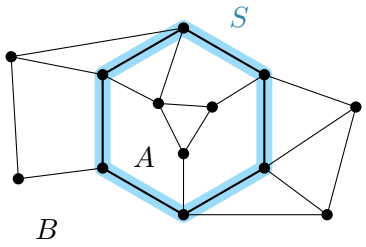


Figure 1: **Jordan curve theorem.**

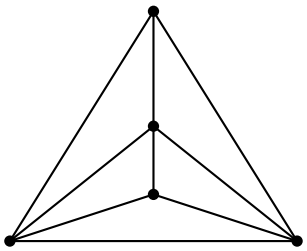


Figure 2: **Triangulated planar graph with $n = 5$.**

Add edges to G to form G' until G' is triangulated. By the property above, G' remains planar and any sets A, B, and S on G' work for G. Run a breadth-first search (BFS) from any root node ρ in G' and call the BFS tree T. Let $L(i)$ be the set of nodes that are at level i in T, and observe that any edge in G' connects nodes within the same level or two consecutive levels, so removing a level disconnects the levels before it from the levels after it.

Let l_1 indicate the first BFS level in T such that levels $0, 1, \ldots, l_1$ contain at least $\frac{n}{2}$ nodes. Therefore, the levels before l_1 contain fewer than $\frac{n}{2}$ nodes and the levels after contain at most $\frac{n}{2}$ nodes. From here, we will choose an $l_0 < l_1$ and an $l_2 > l_1$ such that these BFS levels separate G'

into a low and high section corresponding to one set, a middle section corresponding to another, and the l_0 and l_2 that will make up the cut S.

Let $s = \lceil \sqrt{n} \rceil$. All levels together contain at most n nodes, so by the pigeonhole principle, within s levels of l_1, there exists a level containing at most $\sqrt{n}$ nodes. We use a window of size s from l_1 to find a bounded size l_0 and l_2, and for the case where the tree is shallow and this window goes beyond the root level or the last BFS level, we choose $L(l_0) = \{\rho\}$ (the root), or an empty set $L(l_2) = \emptyset$. To summarize,

$$|L(l_0)| \leq \sqrt{n}, \quad |L(l_2)| \leq \sqrt{n}, \quad l_1 - l_0 < s, \quad l_2 - l_1 < s, \quad l_0 < l_1 < l_2.$$

Delete the levels l_0 and l_2, costing at most $2\sqrt{n}$ nodes, and define the bands

1. *Low*: levels before l_0
2. *Mid*: levels between l_0 and l_2
3. *High*: levels after l_2

By the choice of l_1, *Low* has fewer than $\frac{n}{2}$ nodes and *High* has at most $\frac{n}{2}$ nodes, but *Mid* is unbounded.

Case 1: *Mid* has at most $\frac{2n}{3}$ nodes. The two deleted levels are already a separator, so set $S = L(l_0) \cup L(l_2)$ and observe $|S| \leq 2\sqrt{n}$. All three bands contain at most $\frac{2n}{3}$ nodes, so group them into A and B accordingly: if the largest band has at least $\frac{n}{3}$ nodes, set that to A and the remaining bands to B; otherwise all bands have fewer than $\frac{n}{3}$ nodes, so combine the two smallest into A and set the remaining band to B.

Case 2: *Mid* has more than $\frac{2n}{3}$ nodes. *Mid* itself must be cut into smaller pieces. Define G'' to be the graph obtained from G' by contracting all nodes in levels $0, 1, \ldots, l_0$ into one root node ρ^* and removing levels $\geq l_2$. By the window size s separating l_0 and l_2 from l_1, observe that all nodes are close to the root ρ^* and the BFS tree T' on G'' rooted at ρ^* is a shallow spanning tree of maximum depth $2s - 2$.

Now take any edge e of G'' not in T'. Its two endpoints are joined to ρ^* by a tree path in T', and we can join the two paths with e to form a cycle of maximum length $4s - 3$. By the Jordan Curve Theorem, this cycle splits the remaining nodes of *Mid* into two regions, but these regions may not be balanced.

To balance these regions, we use the fact that G'' is triangulated. While one region has more than $\frac{2n}{3}$ nodes, repeatedly add or remove a tangent triangle from the enclosed region. We maintain the bounded cycle size by observing that any enclosed cycle here is still two tree paths plus an enclosing edge. When both regions are bounded by $\frac{2n}{3}$ nodes, set A to be the larger region, B to be the rest of *Mid* plus all nodes of *Low* and *High*, and set S to be the cycle in *Mid* as well as $L(l_0)$ and $L(l_2)$ of G'. □

5.1.2 Applying the Lipton–Tarjan Planar Separator Theorem

Theorem 7 (restated). *For all $0 < \varepsilon \leq 1$ and $\Delta > 1$, there exists a constant c such that every planar graph of maximum degree $\leq \Delta$ is $(\varepsilon\Delta, \frac{c}{\varepsilon^2})$-hyperfinite.*

Proof. **Goal**: Remove at most $\varepsilon\Delta n$ edges such that every connected component has at most $k = \frac{c}{\varepsilon^2}$ nodes.

Repeatedly use the planar separator theorem (Lemma 9) until all connected components have at most k nodes. Each use splits a graph into two sides of at most $\frac{2h}{3}$ nodes at a cost of $7\sqrt{h}$ nodes, where h is the size of the connected component separated.

The total cost of the nodes deleted is shown as a geometric series on groups of pieces that fall into similarly sized buckets.

1. Take all pieces containing between b and $\frac{3b}{2}$ nodes. A cut inside any of these pieces is guaranteed to fall outside of this range because a cut piece is at most $\frac{2}{3}$ as large. Therefore, each node can only be a part of this bucket once and the bucket's size sums to at most n.

2. A piece with $t \geq b$ nodes costs at most $7\sqrt{t} = \frac{7t}{\sqrt{t}} \leq \frac{7t}{\sqrt{b}}$ nodes. Summing over all pieces in the bucket, the total cost is $\leq \frac{7n}{\sqrt{b}}$.

3. The buckets are $b = k, \frac{3k}{2}, (\frac{3}{2})^2 k, \ldots, n$, whose upper bound costs follow a geometric series.

$$\sum_{j=0}^{J} \frac{7n}{\sqrt{k(3/2)^j}} = \frac{7n}{\sqrt{k}} \sum_{j=0}^{J} \left(\sqrt{\frac{2}{3}}\right)^j \leq \frac{7n}{\sqrt{k}} \cdot \frac{1}{1 - \sqrt{2/3}}$$

Thus, cutting a planar graph into pieces of size at most k costs at most $\frac{c_1 n}{\sqrt{k}}$ nodes. By the definition of hyperfinite graphs, we can afford εn nodes. Setting these equal and solving for k gives $k = \frac{c_1^2}{\varepsilon^2}$, which reduces to some constant c such that $k = \frac{c}{\varepsilon^2}$. Lastly, since we deleted nodes with the planar separator theorem, we need to convert nodes to edges. Since the graphs we are working on have degree at most Δ, at most εn deleted nodes implies at most $\varepsilon\Delta n$ deleted edges. Finally, by repeatedly applying the planar separator theorem to remove at most $\varepsilon\Delta n$ edges to leave components of maximum size $k = \frac{c}{\varepsilon^2}$ nodes, we have shown that degree-bounded planar graphs are $(\varepsilon\Delta, \frac{c}{\varepsilon^2})$-hyperfinite. □