A Necessary Dose of Mathematics for Suffix Trees

Phillip Compeau

Spring 2019

1. A Little About Graphs

First, some notation.

- Denote the nodes (a.k.a. vertices) of a graph G as $V(G)$, and its edges as $E(G)$.
- Denote the degree of a node v by $deg(v)$.
- We use $|S|$ denote the number of elements in a set S.

We will first prove a statement about graphs in general.

Proposition 1. $2|E(G)| = \sum_{v \in V(G)} deg(v)$.

Proof. Consider counting the number of edges incident to each node. When we do this, each edge $\{v, w\}$ is counted exactly twice; once when we count the degree of v, and once for w. □

This lemma exemplifies a classic paradigm in combinatorics (the mathematics of counting), in which we show that two quantities are the same by showing that one process of counting things is equal to both of them.

2. A Little About Trees

A **tree** is defined a connected (undirected) graph without cycles. Trees have **leaves**, which have degree 1, and **internal nodes**, which have degree larger than 1. They may also have a **root** (note that the root of a suffix tree could have degree 1 or larger), so we consider the root as a third class of node. We use $Leaves(T)$ and $Int(T)$ to denote the leaves and internal nodes of a tree, respectively.

As we proved a statement about graphs, we will prove one about trees.

Proposition 2. *Every tree with n nodes has exactly $n - 1$ edges.*

The proof of this statement, which proceeds below, will use the following lemma, which we leave as an exercise.

Lemma 3. *The removal of an edge from a tree produces two trees.*

Proof. We proceed by induction. The only tree with 1 node has 0 edges. Assume that the statement holds for all trees with fewer than n nodes, and consider an arbitrary tree T. If we remove an arbitrary edge from T, then by the preceding lemma, we obtain two trees with j and k nodes. By the inductive hypothesis, these trees have $j - 1$ and $k - 1$ edges. Therefore, we know that T must have $(j - 1) + (k - 1) + 1 = (j + k) - 1 = n - 1$ edges. □

3. Suffix Trees Take Linear Space

Let *SuffixTree*(*Text*) denote the suffix tree of *Text*. Because all maximal non-branching paths in the suffix trie of *Text* were compressed into edges to form *SuffixTree*(*Text*), no nodes other than the root have degree 2 (otherwise, they would have been compressed).

In class, we established that *SuffixTree*(*Text*) has $|Text| + 1$ leaves. Note that all the nodes of *SuffixTree*(*Text*) have degree at most equal to one more than the size of the alphabet . These two facts, taken with the following theorem, establish that *SuffixTree*(*Text*) requires $O(Text)$ storage.

Theorem 4. *SuffixTree*(*Text*) *has at most* $|Text| - 1$ *internal nodes (and therefore at most* $2|Text| + 1$ *total nodes).*

Proof. Let T denote *SuffixTree*(*Text*) for short. By Proposition 1, we know that

$$2|E(T)| = \sum_{v \in V(T)} deg(v).$$

Applying Proposition 2 yields

$$2(|V(T)| - 1) = \sum_{v \in V(T)} deg(v).$$

When we split each side into leaves, internal nodes, and the root, we obtain

$$2(|Leaves(T)| + |Int(T)| + 1 - 1) = \sum_{v \in Leaves(T)} deg(v) + \sum_{v \in Int(T)} deg(v) + 1.$$

The first summation is equal to $|Leaves(T)|$. Because all internal nodes have degree at least 3, the second sum is at least $3|Int(T)|$. Thus,

$$2|Leaves(T)| + 2|Int(T)| \geq |Leaves(T)| + 3|Int(T)| + 1.$$

Remember that $|Leaves(T)|$ is equal to $|Text| + 1$. We therefore obtain

$$2|Text| + 2|Int(T)| \geq |Text| + 3|Int(T)| + 1,$$

and algebra yields

$$|Text| - 1 \geq |Int(T)|,$$

which is what we set out to prove. □