CSE 421 Lecture 3

1 Bounding m for Trees

Last lecture we proved that.

Lemma 1. *If G has no cycle, then it has a vertex of degree at most* 1.

Now, we finish the result:

Theorem 2. *Every tree with n vertices has exactly $n-1$ edges.*

Proof:

Let $P(n)$ be the statement "Every tree with n vertices has exactly $n-1$ edges".
We prove by induction on n.

Base $n = 1$:
Any graph with 1 vertex has no edge.

Induction:
Let T be a tree with n vertices.
By Lemma 1, there is a vertex $v \in T$ with degree 1 (Can't be degree 0 because T is connected).
Remove v to create a new graph T'.
Note that
T' has no cycle (because $T' \subset T$ and T has no cycle)
T' is connected (because we removed only degree 1 vertex).
Hence, T' is a tree with $n-1$ vertices.
By induction, T' has $n-2$ edges. Hence, T has $n-1$ edges.

2 Bounding m for Planar Graphs

Definition 3. A graph is planar if it can be drawn on the plane in such a way that its edges intersect only at their endpoints. A face is a region bounded by edges, including the outer, infinitely large region.

Lemma 4 (Euler's Formula)**.** *Given a connected planar graph with v vertices, e edges and f faces. We have*

$$f + v = e + 2.$$

Proof:
Let $P(e)$ be the statement "$f + v = e + 2$".
We prove by induction on e.

Base $e = 0$:
The graph has 1 vertex and 1 face. Hence, $1 + 1 = 0 + 2$.

Induction:
Case 1) G does not has a cycle.
G is a tree. We know $f = 1$ and $e = v - 1$. Hence, $f + v = v + 1 = e + 2$.

Case 2) G has a cycle.

Pick an edge p on a cycle. Remove p to create a new graph G'.

Since the cycle separates the plane into two regions, the regions to either side of p must be distinct. When we remove the edge p, we merge these two regions. So G' has one fewer faces than G.

Let f', e', v' be the number of faces, edges and vertices in G'.

We have $f' = f - 1$, $e' = e - 1$ and $v' = v$.

Since $f' + v' = e' + 2$ (by induction), we have $f + v = e + 2$.

Theorem 5. *Given a planar graph with v vertices. Suppose that $v \geq 3$, then it has at most $e \leq 3v - 6$ edges.*

Proof:

If the planar graph is not connected, we can prove it for each connected components and sum up the inequalities. Hence, we can assume the planar graph is connected.

We define the degree of a face to be the number of edges enclosing the face.

Since each edge touches two faces, we have

$$\sum_{\text{faces } f} \deg(f) = 2e.$$

Since each face must have degree at least 3, we have $3f \leq 2e$.

By Euler's Formula, we have

$$f = e - v + 2.$$

Hence

$$2e \geq 3f = 3e - 3v + 6.$$

This gives $e \leq 3v - 6$.

3 BFS Tree

Lemma 6. *All non-tree edges connects vertices on the same or adjacent levels of the tree.*

Proof:

Consider an edge xy.

Assume x is first discovered at level i.

Then, we know $L[y] \geq i$.

When we discover x, we set all undiscovered neighbors of x to level $i + 1$.

Hence, $L[y] \leq i + 1$.

Theorem 7. *(BFS Tree gives you the shortest path information). Level i in the tree are exactly all vertices with $d_G(s, v) = i$ where s is the starting vertex and $d_G(s, v)$ is the shortest path distance from s to v on G.*

Proof:

Fix any vertex v.

1) $d_G(s, v) \leq L(v)$

The BFS Tree gives a path of length $L(v)$ from s to v.

2) $d_G(s, v) \geq L(v)$

Suppose the shortest path length is i. Say $s = v_0, v_1, \cdots, v_i = v$ is the shortest path.

By previous Lemma, we know

$$\begin{aligned} L(v_0) &= L(s) = 0, \\ L(v_1) &\leq L(v_0) + 1 \\ L(v_2) &\leq L(v_1) + 1 \\ &\vdots \\ L(v_i) &\leq L(v_{i-1}) + 1. \end{aligned}$$

Hence, $L(v) = L(v_i) \leq i = d_G(s, v)$.

4 Quiz: Finding Cycle

Problem 8. Give an algorithm to detect whether a given undirected graph contains a cycle.

Theorem 9. *We can detect whether a given undirected graph contains a cycle in $O(m + n)$ time.*

Proof:
Since we can test existence of cycle separately in each connected component, we assume the graph is connected.

Algorithm:

- Run BFS starting at any vertex.
- Let T be the BFS tree.
- If $G = T$,
  - Output no cycle.
- Else
  - Let $e = (v, w)$ be an edge in G but not in T.
  - Let u be the least common ancestor of v and w in T.
  - Output the cycle $v \to u \to w \to v$.

.

Runtime:
$O(m + n)$. The bottleneck is BFS.

Correctness:

Case 1) $G = T$.
Tree has no cycle

Case 2) $G \neq T$
Since $T \subset G$, there is an edge $e \in G$, but not in T. The algorithm uses e to construct a cycle.