Trees and Rooted Trees

1 The number of leaves in a tree

What is the minimum number of leaves in a tree with at least 2 vertices

1.1 Lemma

A tree with at least 2 vertices, n_3 of which have degree at least 3, has at least $n_3 + 2$ leaves

1.2 Proof

Let T be a tree on $n \geqslant 2$ vertices. We use induction on n

- Let $l(T)$ denote the number of leaves in T, and $n_3(T)$ denote the number of vertices of degree at least 3 in T
- Induction base: If n=2, then n_3=0 and T has 2 leaves
- Step: Now suppose that every tree on $< n$ vertices has at least $n_3 + 2$ leaves (induction hypothesis), and consider a tree T on $n \geqslant 3$ vertices
- Since T is a tree on at least 3 vertices, T has a leaf u
- Then T′=T-u is a tree on n-1 vertices. By the induction hypothesis we have $l(T') \geqslant n_3(T') + 2$
- We have: a leaf u in T, a tree T′=T-u, $l(T') \geqslant n_3(T') + 2$
- Let v be the (unique) neighbour of u in T
- T is connected and has at least 3 vertices, so v has at least 2 neightbours in T
- The rest of the proof is by **case analysis**

1. Suppose that v has exactly 2 neighbours in T
   - Then $n_3(T') = n_3(T)$ and $\ell(T') = \ell(T)$
   - Hence, $\ell(T) = \ell(T') \geq n_3(T') + 2 = n_3(T) + 2$
2. Suppose that v has exactly 3 neighbours in T
   - Then $n_3(T') = n_3(T) - 1$ and $\ell(T') = \ell(T) - 1$
   - Hence, $\ell(T) = \ell(T') + 1 \geq n_3(T') + 2 + 1 = n_3(T) - 1 + 2 + 1 = n_3(T) + 2$
3. Suppose that v has at least four neighbours in T
   - Then, $n_3(T) = n_3(T')$ and $\ell(T') = \ell(T) - 1$
   - Hence, $\ell(T) = \ell(T') + 1 \geq n_3(T') + 2 + 1 = n_3(T) + 2 + 1 \geq n_3(T) + 2$

This finishes the proof

2 Every tree is a bipartite graph

2.1 Theorem

Every tree is a bipartite graph

2.2 Proof

We give a **direct** proof. We can use the known result on unique paths in a tree T to define a bipartition of its vertex set V(T)

- Choose any vertex v and put this vertex in the set V_1
- For every vertex $u \neq v$, there is a unique path from v to u in T, consider the length of this path
- If the length is odd, put u in V_2, otherwise put u in V_1
- We have to show that this is a valid bipartition
- V_1 and V_2 are disjoint and together make up $V(T)$
- Every edge has end vertices in both V_1 and V_2
- This completes the proof

3 How to find and write down proofs?

These are the questions to ask yourself to help finding a possible proof approach:

- What do I have to **prove**? Is it one statement, or several; is it an implication or an equivalence; can I repharase it; does it resemble other statements?
- What do I **know**? What are the assumptions; do I know the relevant definitions; is there any known theory related to the statement
- Can I get more **insight**? Can I sketch the situation, the assumptions, the question; are there special (small) cases to check; can I break it into several subcases?
- How to **approach/attack** the question? Can I use induction; does a direct proof have any chance; or does it help to use contraposition, or a proof by contradiction?
- Is my solution **valid and convincing**? Write a draft first; check all the steps; critically examine the steps for errors or counterexamples; modify and revise the solution and write it down in a clear way

3.1 The start: write down what you see

We will consider the process of finding the proof on the following example:

3.1.1 Lemma

Let T be a tree on $n \geqslant 2$ vertices, and let $e \in E(T)$. The T-e is a forest consisting of precisely two trees.

3.1.2 Proof

- Clearly, you have to know what a **tree** is, what a **forest** is, and what the **notations** $e \in E(T)$ and $T - e$ mean
- In fact, you have to prove **two**(or perhaps even **three**) statements: T-e is a forest, and this forest consists of precisely two trees (so not $\leqslant 1$ and not $\geqslant 3$ trees)
- Here it (probably) helps to **draw a picture** that roughly sketches the situation and concepts
- If you draw the general situation, and know the definitions and notations, then you more or less **see the solution** in the picture
- The question is **how to write it down** (and check that the picture did not fool you)
- This requires certain **skills and experience**
- You can only learn this by **doing it yourself**
- A **tree** is a connected graph without cycles

- A forest is a graph without cycles
- Sine a tree is a connected graph, between any two vertices there is a path in a tree
- We know from the previous lecture that this path is **unique**

How to use (some of) the above facts to prove that T-e is a forest containing precisely two trees?
Let us consider the first part of the statement first. Can we prove that T-e is a forest

There is an easy consequence of the definitions and so the observation that removing edges from a tree, we cannot introduce cycles. So if T is a tree, then T contains no cycles and T-e contains no cycles either, so T-e is a forest (This is a **direct proof**)

It remains to show that T-e consists of precisely 2 trees, i.e., at least 2 and at most 2 trees. How to prove this?

At least 2: you have to show that T-e is not connected (not 1 tree). This is easy: if u and v are the end vertices of the edge e, then in T-e there is no path between u and v (This is also a direct proof)

At most 2: you have to show that T-e does not consist of 3 or more trees. This is easy, using the observation that the edge e can only connect 2 trees into one. So, if T-e would consist of 3 or more trees, then T is not connected, a contradiction. (This is a proof by contradiction or contraposition)

The proof seems to be complete. Now you have to write it down and **carefully check** the details

3.2 A solution

3.2.1 Proof

Since T is a tree, T - e has no cycles, so T - e is a forest. Since in T - e there is no path between the two end vertices of e, T - e is not connected, hence T - e consists of at least 2 trees.If T - e consists of at least 3 trees, then T cannot be connected. Hence T - e is a forest consisting of precisely two trees.

There are probably many different correct ways to prove the lemma. For instance for the last part you could use the fact that a tree on n vertices has n-1 edges.

So suppose that T-e consists of trees of $n_1, ..., n_k$ vertices for some integer $k \geqslant 1$. Now we count the number of edges of T in two ways: As T has $n_1 + ... + n_k$ vertices, T has $n_1 + ... + n_k - 1$ edges. On the other hand, T has $(n_1 - 1) + ... + (n_k - 1) + 1$ edges. The two expressions can only be equal if k=2, so T-e consists of precisely 2 trees

4 Full m-ary trees

4.1 Definitions

A rooted tree is called a **m-ary tree** if each vertex has at most m children. It is a **full** m-ary tree if each internal vertex has exactly m children

A (full) 2-ary tree is usually called a (full) **binary tree**

Often, the children of each node are assumed to be orderred

4.2 Lemma

A full m-ary tree with i internal nodes has $n = n \cdot i + 1$ vertices

4.3 Proof

Every node except the root is one of m children of a unique internal vertex

Let ℓ be the number of leaves in a full m-ary tree. Since $n = i + \ell$ and $n = m \cdot i + 1$, if we know any of n, i, ℓ then we can find all of them

5 The height of a rooted tree

5.1 Definitions

In a rooted tree, the **level** of a vertex u is the length of the (unique) path from the root to u. (The level of the root is 0)
The **height** of a rooted tree is the maximum level of a vertex in it

5.2 Theorem

There are at most m^h leaves in a m-ary tree of height h

5.3 Proof

Induction on the height h

- Base: If h=1 then the claim is obvious
- Step: Assume the claim is true for m-ary trees of height at most h-1
- Take an m-ary tree T of height $h \geqslant 2$, with root r
- Consider the subtrees of T rooted at children r
- There are at most m of them, and, by induction hypothesis, each has at most m^{h-1} leaves
- Hence, T has at most $m \cdot m^{h-1} = m^h$ leaves

6 Balanced m-ary trees

6.1 Definition

An m-ary tree of height h is **balanced** if all leaves in it have height h-1 or h

6.2 Theorem

If an m-ary tree of height h has ℓ leaves then $h \geqslant \lceil log_m \ell$
If the tree is full and balances then $h = \lceil \log_m \ell \rceil$

6.3 Proof

- The first part immediately follows from the previous theorem: We know that $\ell \leqslant m^h$, so $h \geqslant \log_m \ell$. Since h is an integer, $h \geqslant \lceil \log_m \ell \rceil$
- For the second part, note that there is at least one leaf of level h
- It follows that there are at least m^{h-1} leaves
- So, we have $m^{h-1} < \ell \leqslant m^h$, or taking logarithm to the base m, $h - 1 < \log_m \ell \leqslant h$
- Since h is an integer, $h = \lceil \log_m \ell \rceil$

7 Constructing trees

Every tree $T \neq K_1$ has a leaf. We know that T-v is also a tree. This shows that T can be constructed from a smaller tree T'=T-v by adding a vertex to T' and joining it by one edge to a vertex in T'. This also proves the following statement

7.1 Lemma

We can construct all different trees on $n \geqslant 2$ vertices from all trees on n-1 vertices, by adding one vertex and joining it by one edge to a vertex in one of the trees, in all possible ways, and deleting multiple copies of the same trees

- We can use the above result and procedure to obtain all different trees on n vertices, starting with K_1 (or we can give a **recursive definition** for the class of all trees)
- Check that there are, respectively, 1,1,1,2,3 and 6 different trees on 1,2,3,4,5 and 6 vertices