Announcements

- Homework 2 online, due Sunday
- Homework 1 solutions on course webpage
- Feedback survey on Canvas

Reminder: *DO NOT* submit homework questions to websites (outside of course ones) for help.

And: If you do stumble on Homework solutions online, you must cite them.

Last Time

Minimum Spanning Tree:

- Given a tree with edge weights find a spanning tree with total weight as small as possible.
- Kruskal – Find this by repeatedly adding lightest edge that does not create a cycle

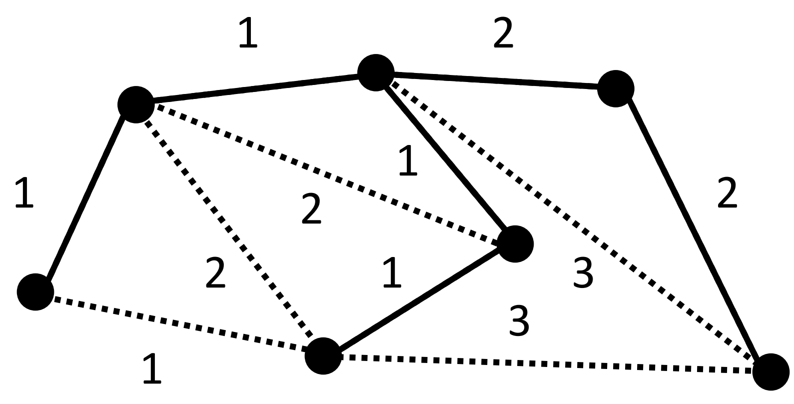
Today

- Prim’s Algorithm for Minimum Spanning Tree
- Counting Trees

Prim’s Algorithm

Another way to find MST:

- Start at base vertex
- Repeatedly add cheapest new edge connected base vertex to something new

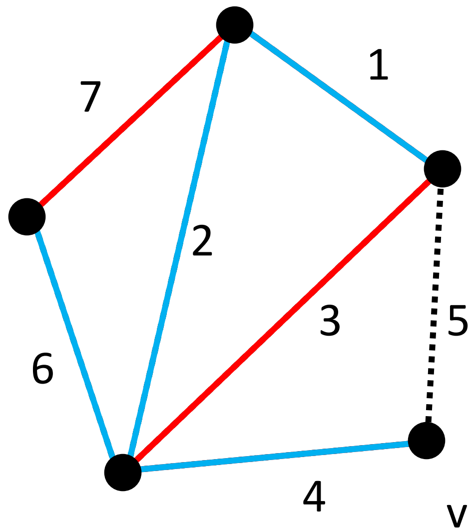


Correctness

Same idea as Kruskal:

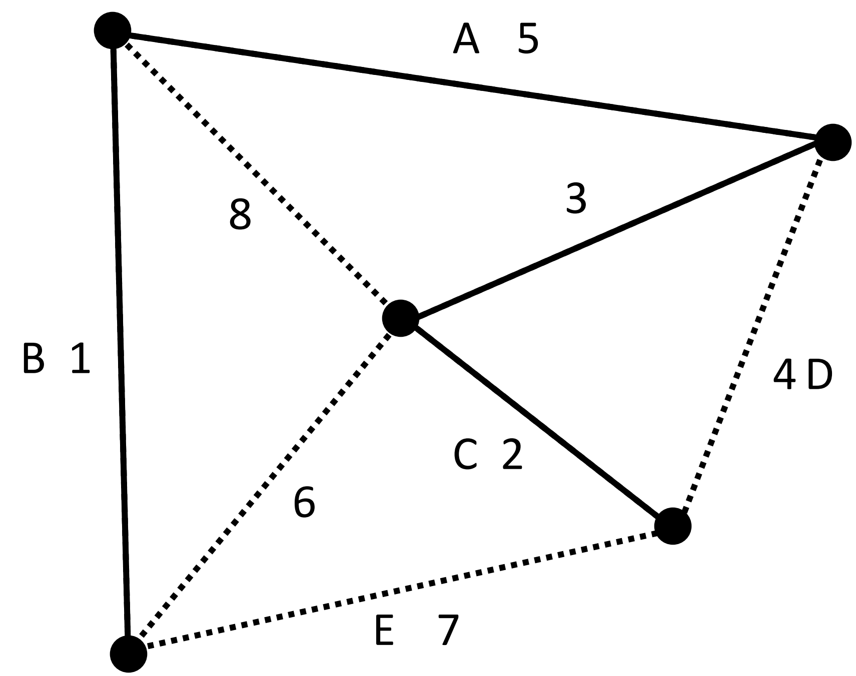
- Take arbitrary tree
- Change to Prim's Tree one edge at a time, each change making things better
- New edge still creates cycle with more expensive old edge, can swap to get new tree

Example



Question: MST

Which of the lettered edges below are in the Minimum Spanning Tree?



Counting Trees

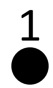
Question: How many trees of size n are there?

It depends a bit on how you count…

Want the number of trees on labeled vertices 1,2,…,n.

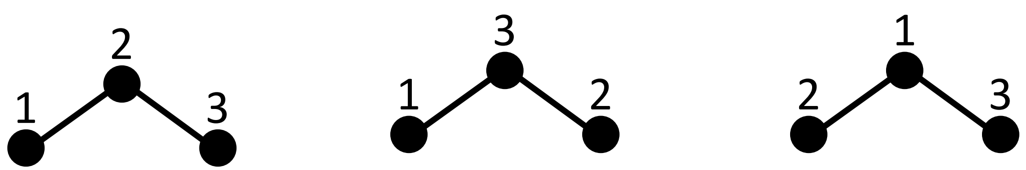
Examples

n=1

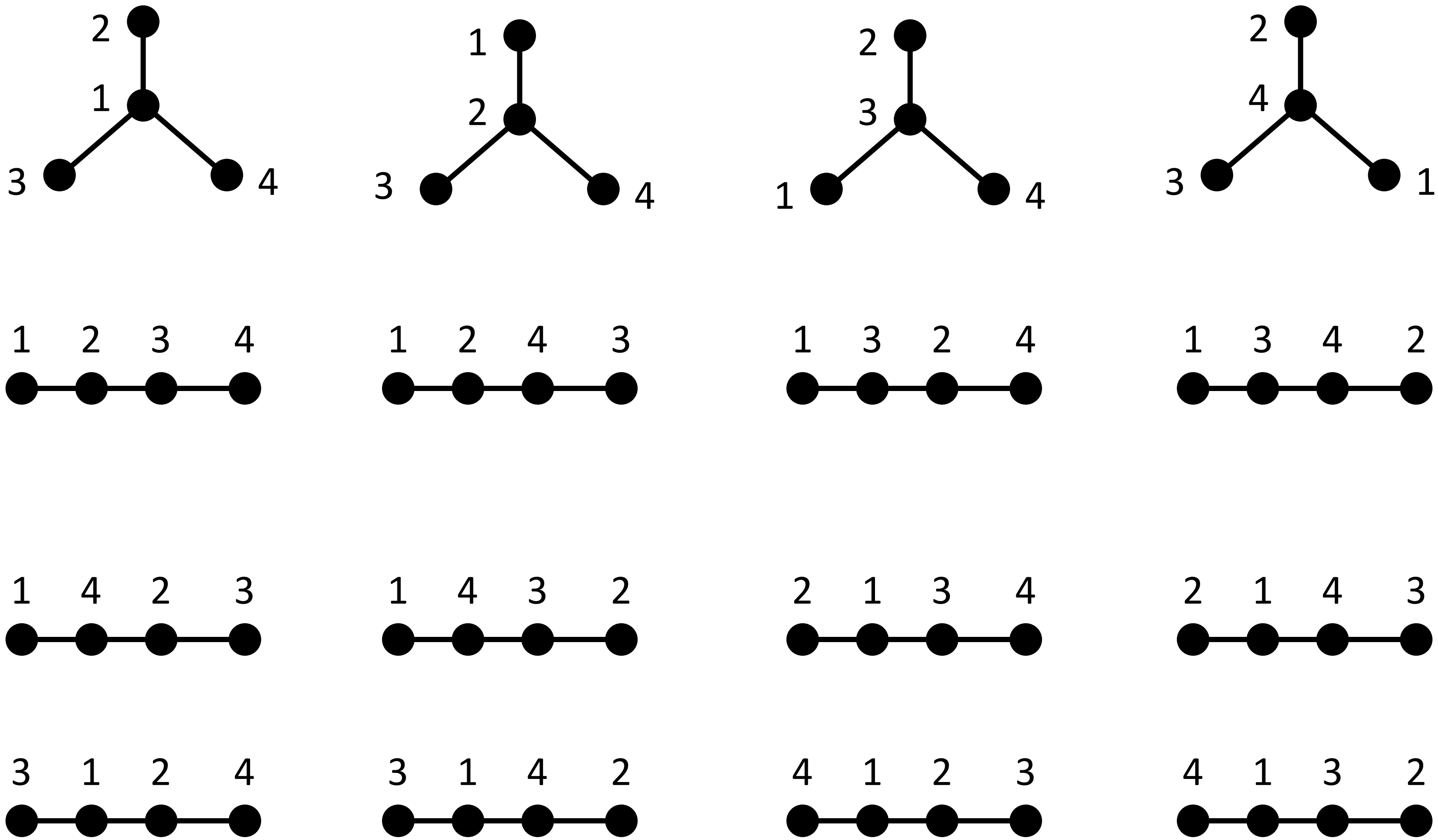


n=2

n=3



n=4


2
1
3
4
1
2
3
4
2
3
1
4
2
4
3
1
1 2 3 4
1 2 4 3
1 3 2 4
1 3 4 2
1 4 2 3
1 4 3 2
2 1 3 4
2 1 4 3
3 1 2 4
3 1 4 2
4 1 2 3
4 1 3 2

Cayley's Theorem

Theorem (1.18): There are n^{n-2} labeled trees of order n.

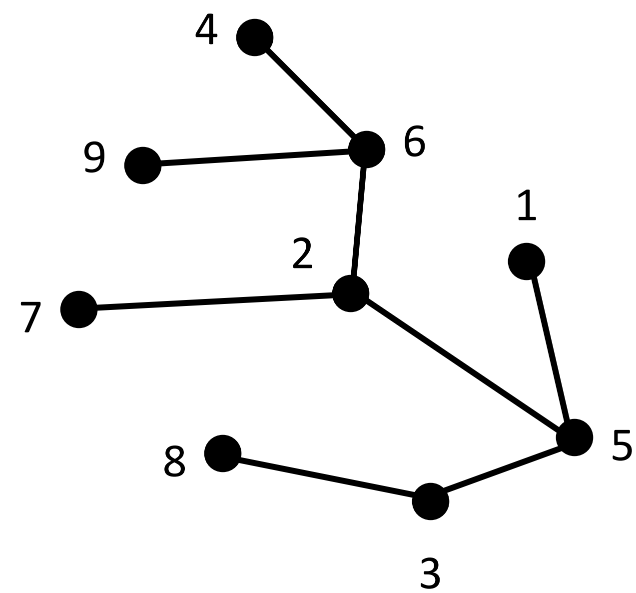
Sequence grows quickly:

1,1,3,16,125,1296,16807,262144, 4782969,100000000…

Proof idea: Find a bijection between trees and sequences of n-2 numbers from 1,2,..,n

How to get a List from a Tree

- Take lowest labeled leaf, v
- Record label of v's neighbor
- Remove v from G
- Repeat until G has only 2 vertices



5, 6, 2, 3, 5, 2, 6

Next Step

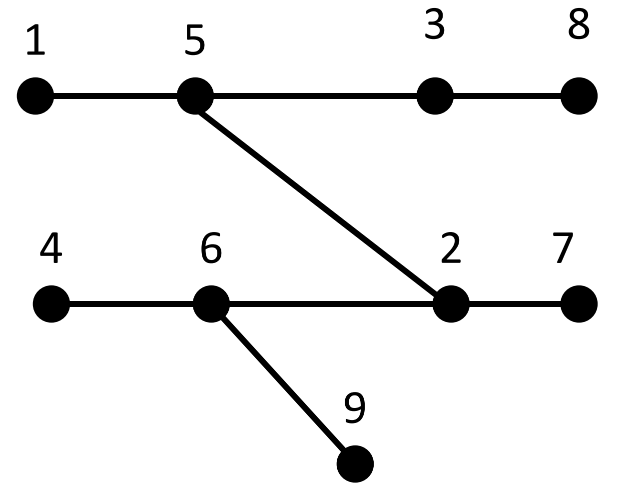
- To every tree on vertices 1,2,…,n assign a list of n-2 numbers from 1 to n.
- There are n^{n-2} many such lists.
- **<u>Need to show:</u>** This is 1-1. Each such list comes from one and only one tree.

How to find the tree

- Find missing numbers. These are the leaves.
- Smallest missing number is v.
- Connect v to first element of list.
- Remove v from available numbers and first element of list
- Repeat until list gone
- Connect remaining elements

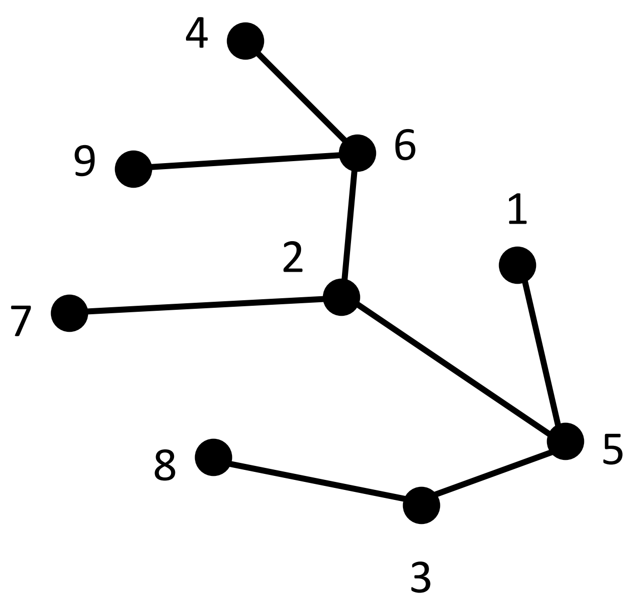
5, 6, 2, 3, 5, 2, 6

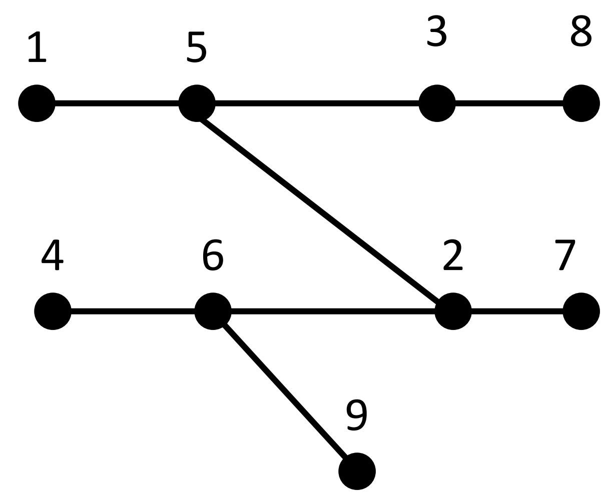
Missing: 1, 4, 7, 8, 9
3, 5, 2,



1, 2, 3, 4, 5, 6, 7, 8, 9

Check





Why does this work?

Produces a tree:

- Each vertex connects to one eliminated later, so no loops.
- Number of edges correct

Produces *same* tree:

- Follows algorithm for getting labels.
- Lowest missing (remaining) number *must* connect to next element of the list.

Generalization: Matrix Tree Theorem

Cayley's Theorem counts the number of spanning trees of K_n. What about other graphs?

Given G compute matrices:

- D – diagonal matrix D_{ii} = deg(i)
- A – adjacency matrix
  - A_{ij} = 1 if i&j adjacent, 0 otherwise
- M = D-A
- M' = M with first row and column removed

Theorem (1.19): #Spanning Trees of G = det(M')