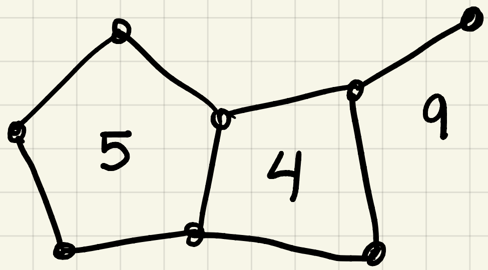
Which graphs are not planar?

First, define degree of a face to be the number of edges on a closed walk of its boundary

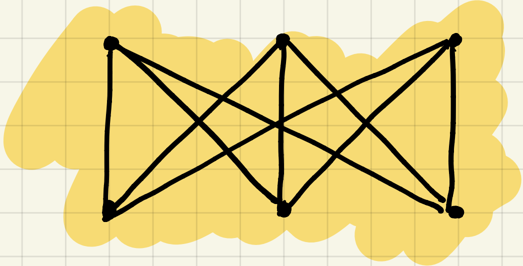


$5+4+9=18$

$e=9$

$$\sum_f d_f = 2e$$

$K_{3,3}$ is not planar

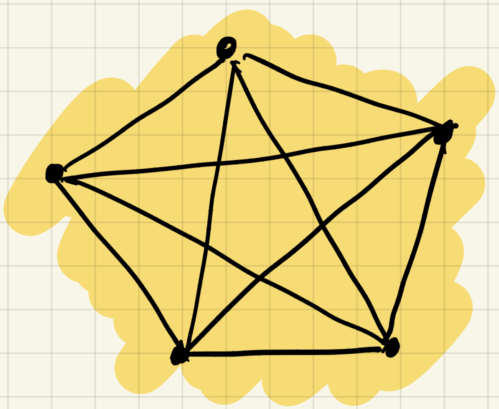


every face has at least 4 edges

$$4f \leq 2e$$

$v=6,\ e=9 \Rightarrow f=5$, but $4\times 5 > 2\times 9$

K_5 is not planar



every face has at least 3 edges

$v=5,\ e=10 \Rightarrow f=7$, but $3\times 7 > 2\times 10$

Every non-planar graph has $K_{3,3}$ or K_5 as a "basic shape"

Another interesting result:

Every planar graph with $v > 2$ satisfies

$$e \leqslant 3v - 6$$

proof: Since every face has degree at least 3 (because $v > 2$)

we have $3f \leqslant 2e$

but $e = v + f - 2 \leqslant v + \frac{2e}{3} - 2 \Rightarrow e \leqslant 3v - 6$

Average degree

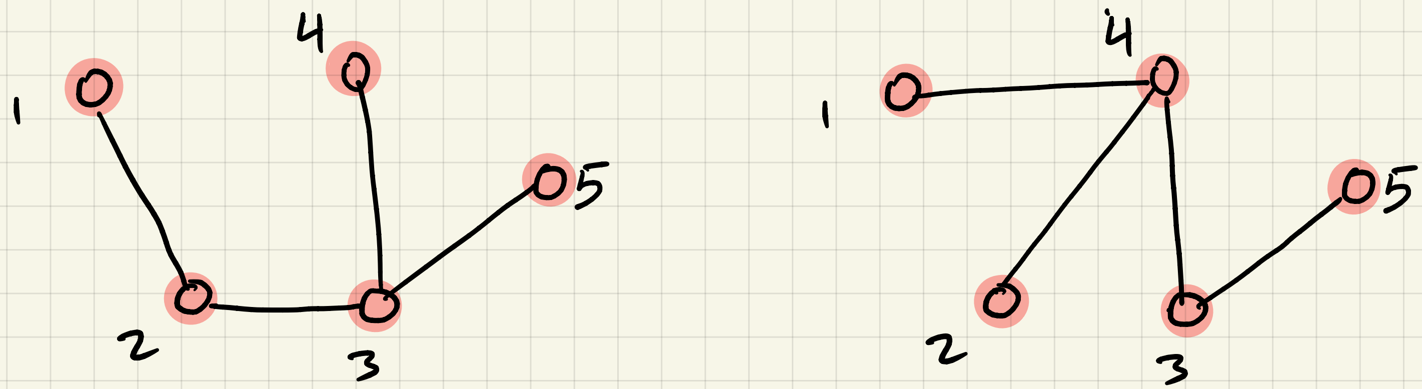
$$\sum_{v \in V} d_v = 2e \leqslant 6v - 12$$

$$\frac{\sum d_v}{v} \leqslant 6 - \frac{12}{v} < 6$$ (there must be a vertex v such that $d_v \leqslant 5$)

Number of Trees

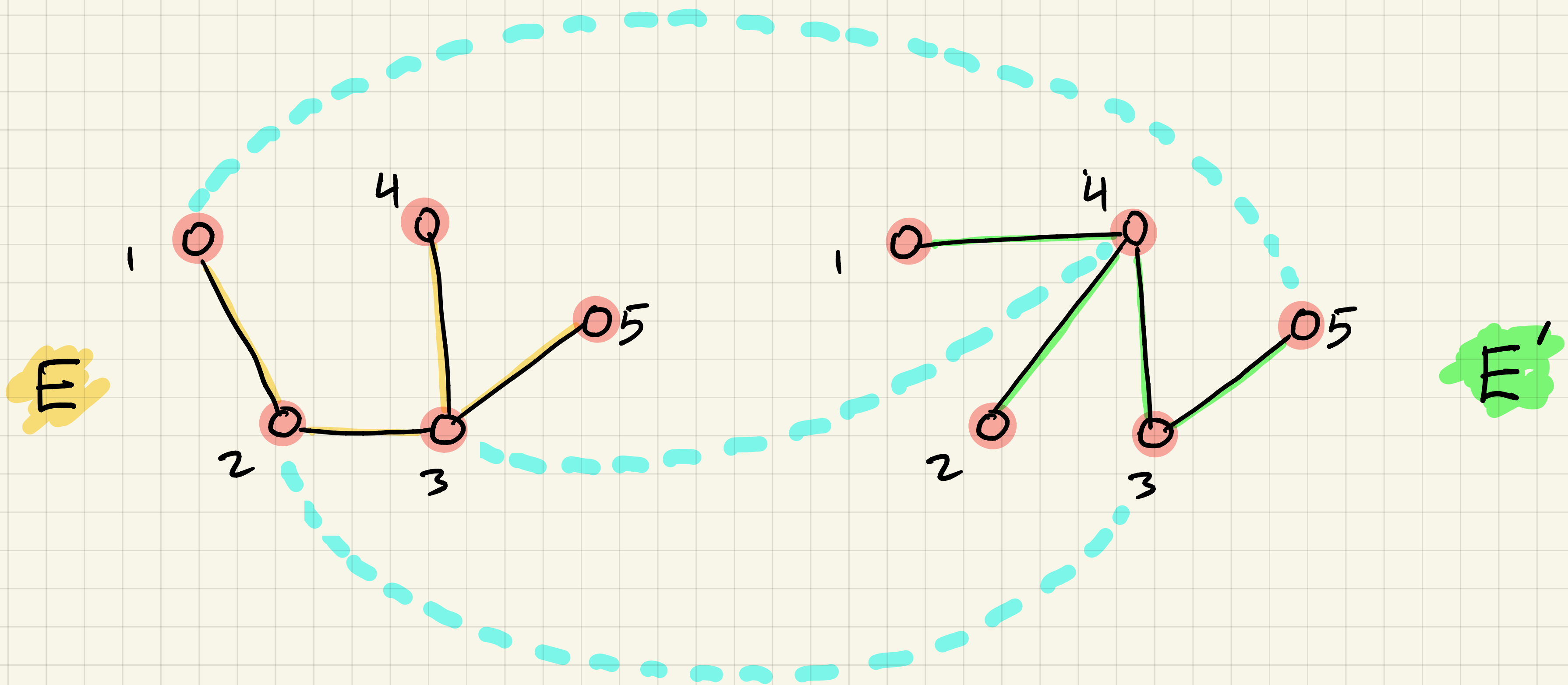
Given n vertices, how many trees can we make?

It depends! Are these the same?



Labeled trees: Two trees are the same if they have the same set of edges

Unlabeled trees: Two trees are the same if there exists a bijection between their vertices that preserves the edges.



$f(1) = 5 \quad f(2) = 3 \quad f(3) = 4 \quad f(4) = 1 \quad f(5) = 2$

$(u, v) \in E \iff (f(u), f(v)) \in E'$

Cayley's Formula:

Number of labeled trees on n vertices is n^{n-2}

Chapter 8 contains 3 proofs

1) Prufer code
2) Inclusion-Exclusion
3) A counting argument

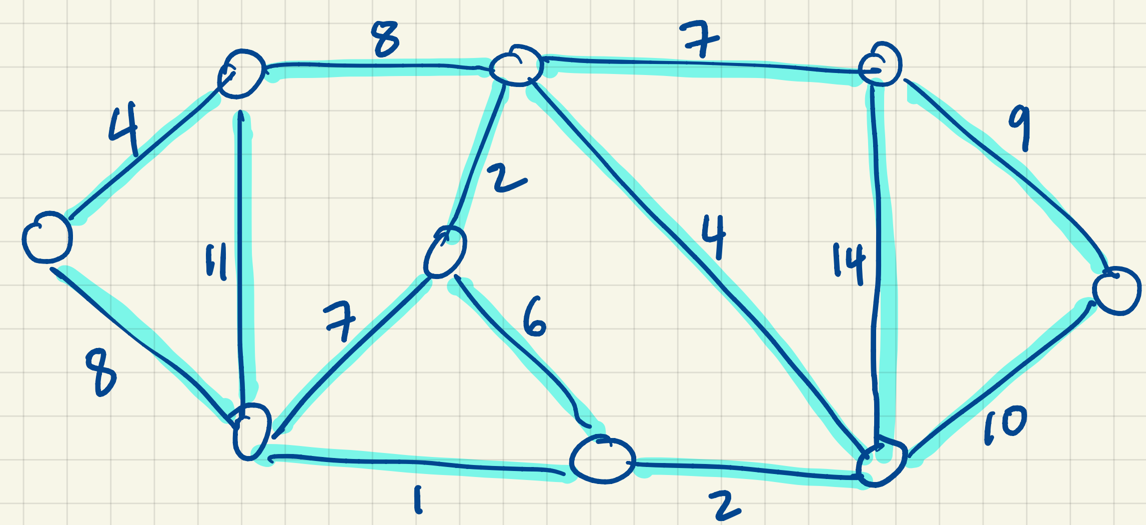
Finding the best tree : The min. Spanning tree

Given a connected weighted graph : a graph with a weight function of the edges

$$w : E \longrightarrow \mathbb{R}$$

Find a tree (connected acyclic) that has the smallest total weight.

Example :



Brute force:

Trying all possible trees is bad: There are many!

An algorithm that works: Greedy algorithm.

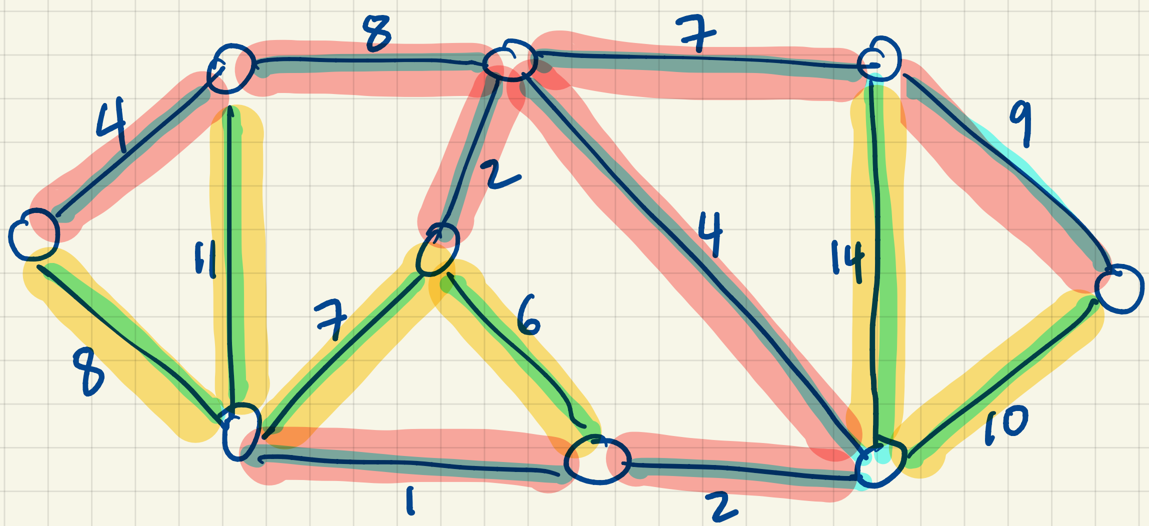
Pick the smallest weight edge and add it to the tree as long as it makes no cycle (so sort the edges by weight and go through them one at a time)

Remember: Sorting is $|E| \log |E|$ time.

Checking for cycles can be done efficiently.

Proof that alg. produces best tree: CSCI 335/Chapter 8

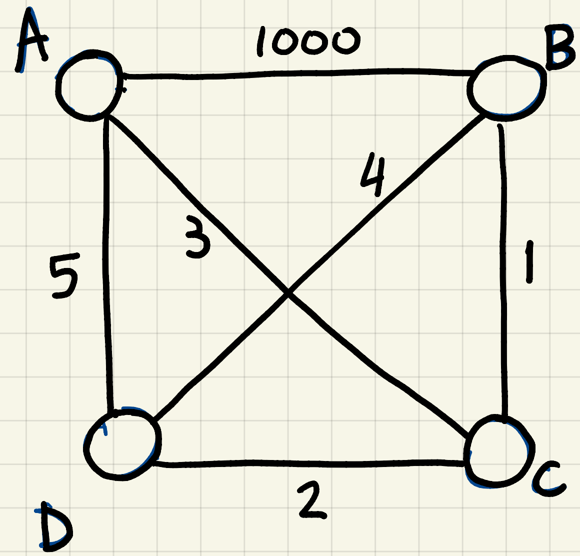
Running the greedy alg. together in class...



weight of tree: $1 + 2 + 2 + 4 + 4 + 7 + 8 + 9 = \ldots$

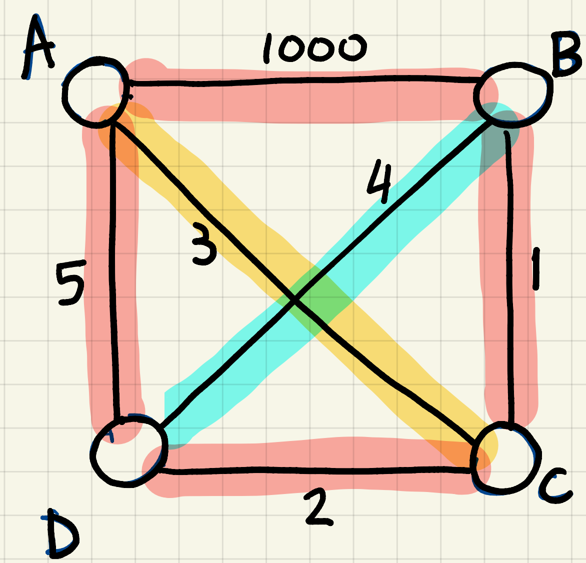
Greedy does not always work!

Example: Traveling salesman: Find a cycle that visits every vertex exactly once at min. cost.



Greedy: Pick the smallest weight edge and add it to cycle as long as degree of every vertex ≤ 2 and cycle does not close early (missing some)

Running the alg. together in class...



optimal cycle has weight $1+3+5+4=13$

cycle found by greedy alg. has weight $1+2+5+1000$

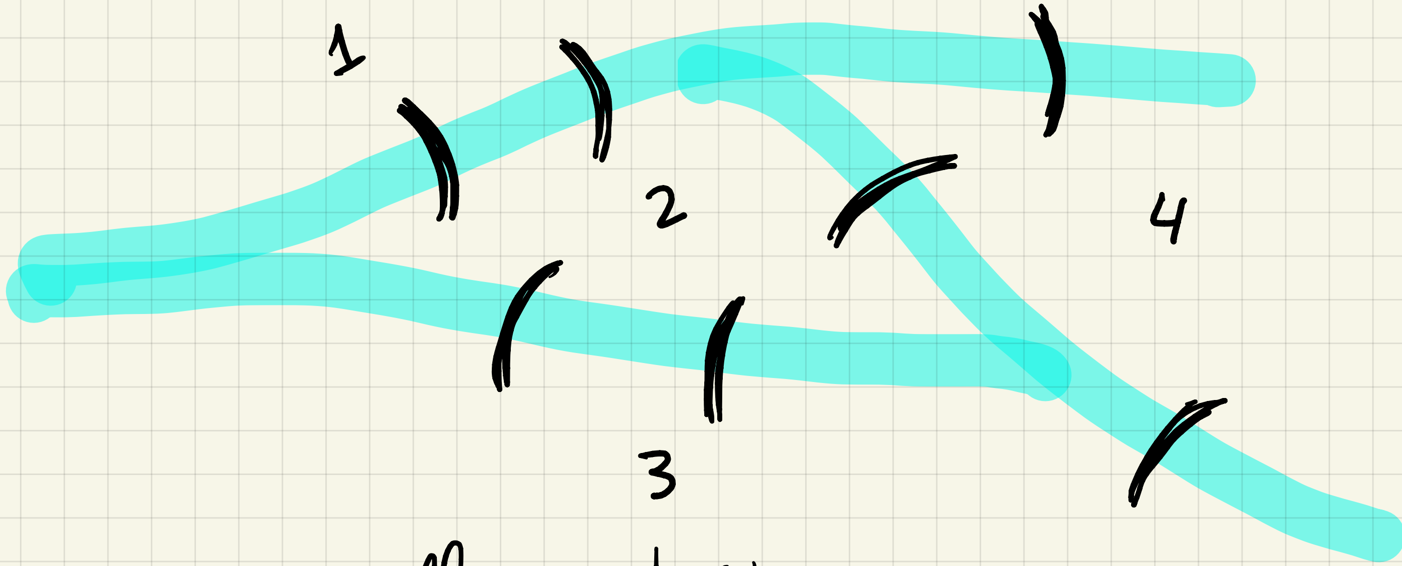
Hamiltonian Cycle: A cycle that visits every vertex exactly once.

Hard to find!

Euler cycle: A cycle that visits every edge exactly once. Easy

(Inspired by Bridges of Konigsberg, now Kaliningrad, Russia)

Euler 1735



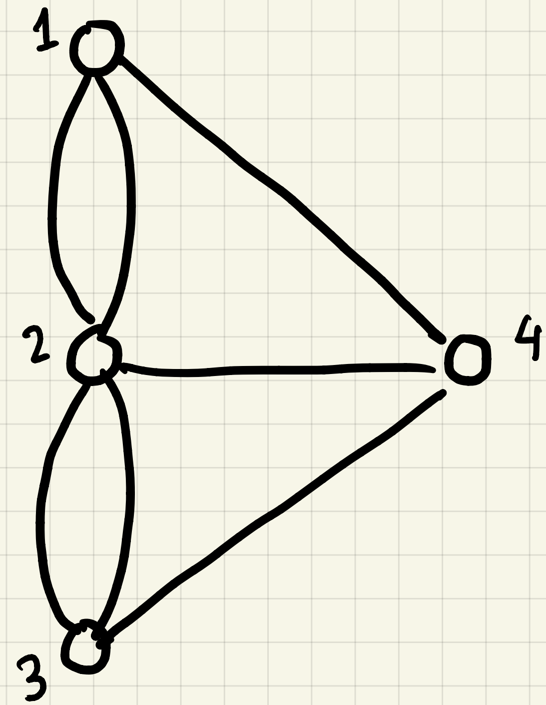
Can we cross all 7 bridges and return to where we start ?

Euler's multigraph :

Graph with multiple edges between vertices

Each bridge is a edge

Find an Euler cycle.



$$\boxed{\begin{array}{l}\text{All vertices have} \\ \quad\text{even degree}\end{array} \iff \text{Euler cycle exists.}}$$

Easy to find one:

Repeat
- Pick some arbitrary vertex
- Follow new edges arbitrarily until you can't (Found a cycle)
- Join with prev. cycle

Until done.

Example: