

## In any other words...

MegaCorp has expanded internationally, and needs to hire translators in a hurry. They found  $m$  people that can translate between pairs of languages, representing a total of  $n$  different languages. Of these, only  $k$  distinct key languages are needed by the company and they need to be able to translate between all combinations of them.

The bottom line is that they want to hire as few translators as possible that, through some chain of translations, can convert between any pair of the key languages. They don't care how long such a chain is, as long as it exists.

# Representing the Problem

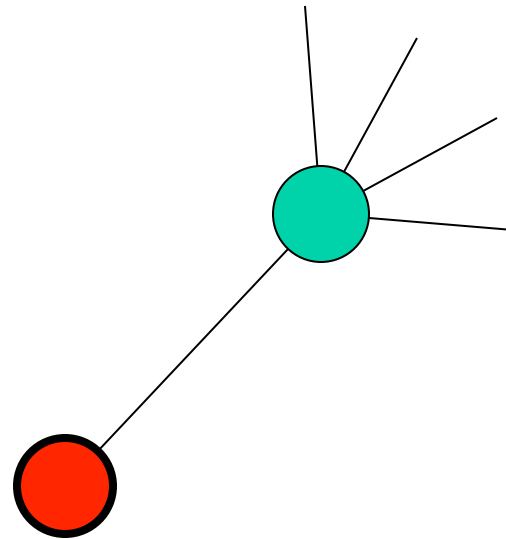
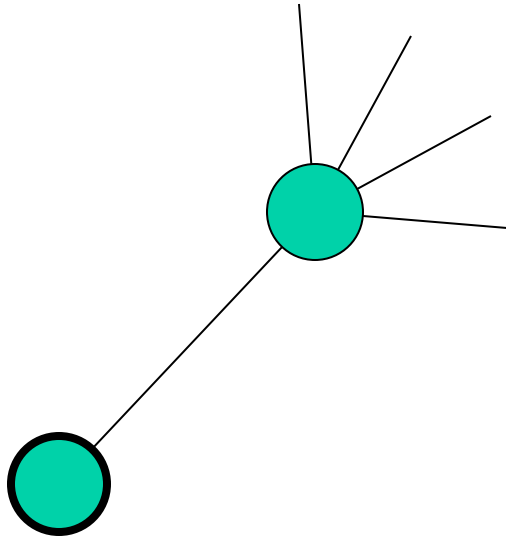
The company represents this problem as a graph where the  $n$  languages are the vertices, and the  $m$  translators are edges connecting pairs of languages. The goal is to find a connected sub-tree of minimum size that contains all  $k$  key vertices.

This boils down to a simple version of the *Steiner tree* problem.

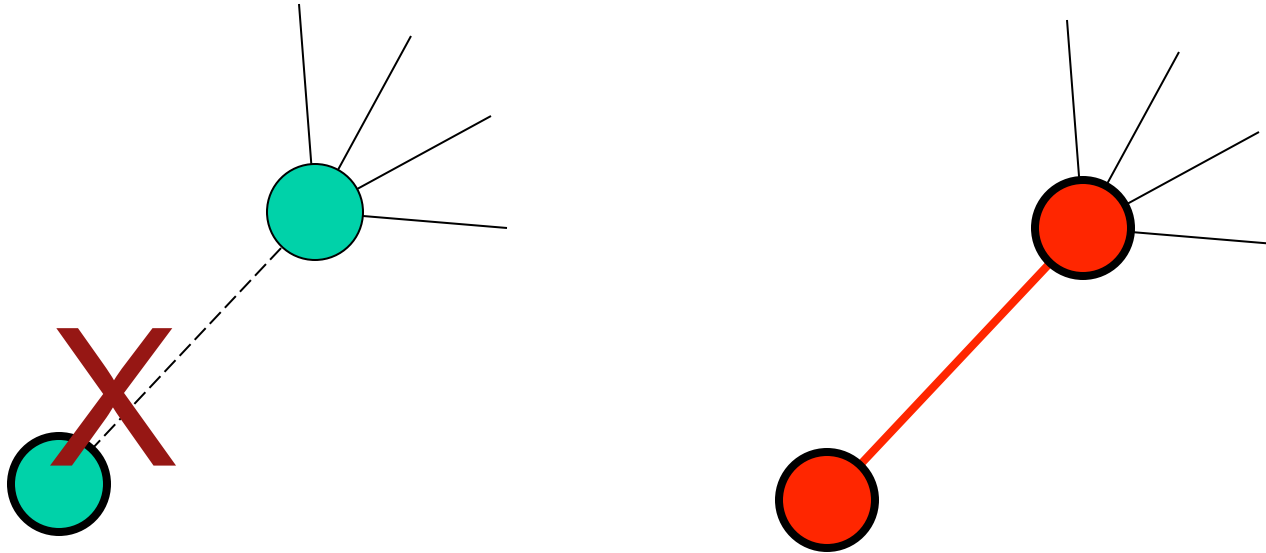
# Questions to Ask...

- What is the brute-force approach?
- How can we set bounds on the solutions?
- When will we need to backtrack from an illegal state?
- What order should we test the potential members in when branching?
- How can we simplify the problem as far as possible in polynomial time?

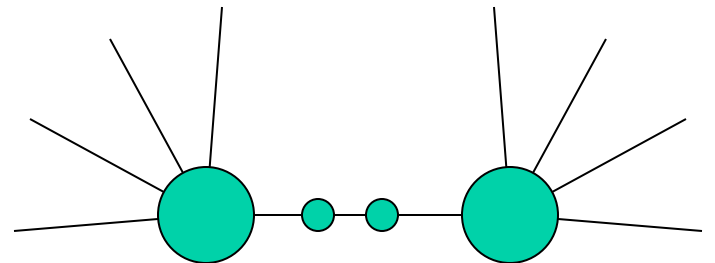
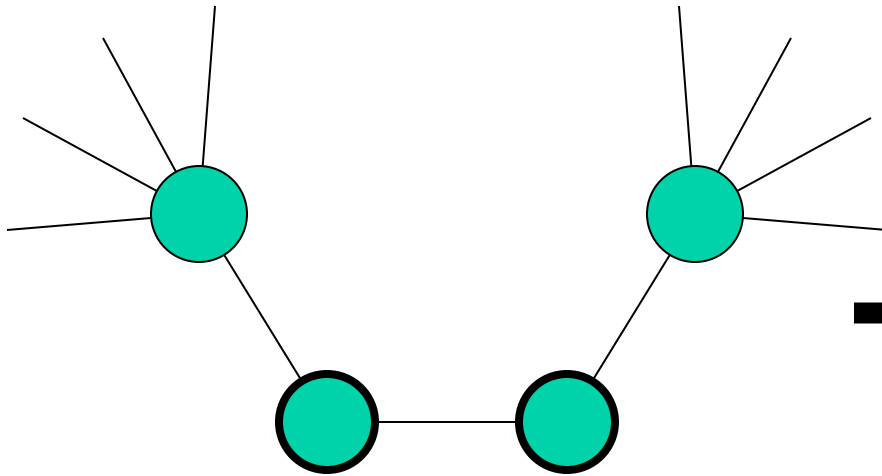
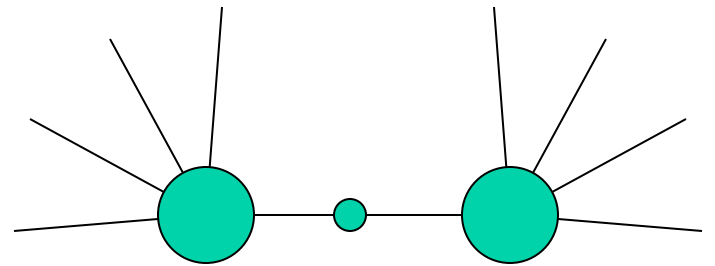
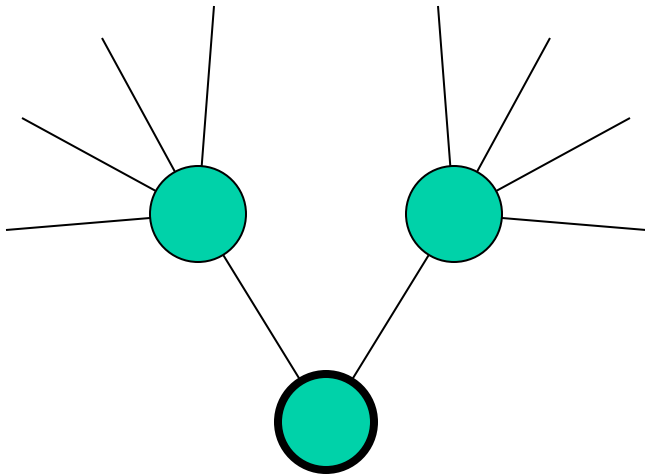
# Degree 1 Vertices



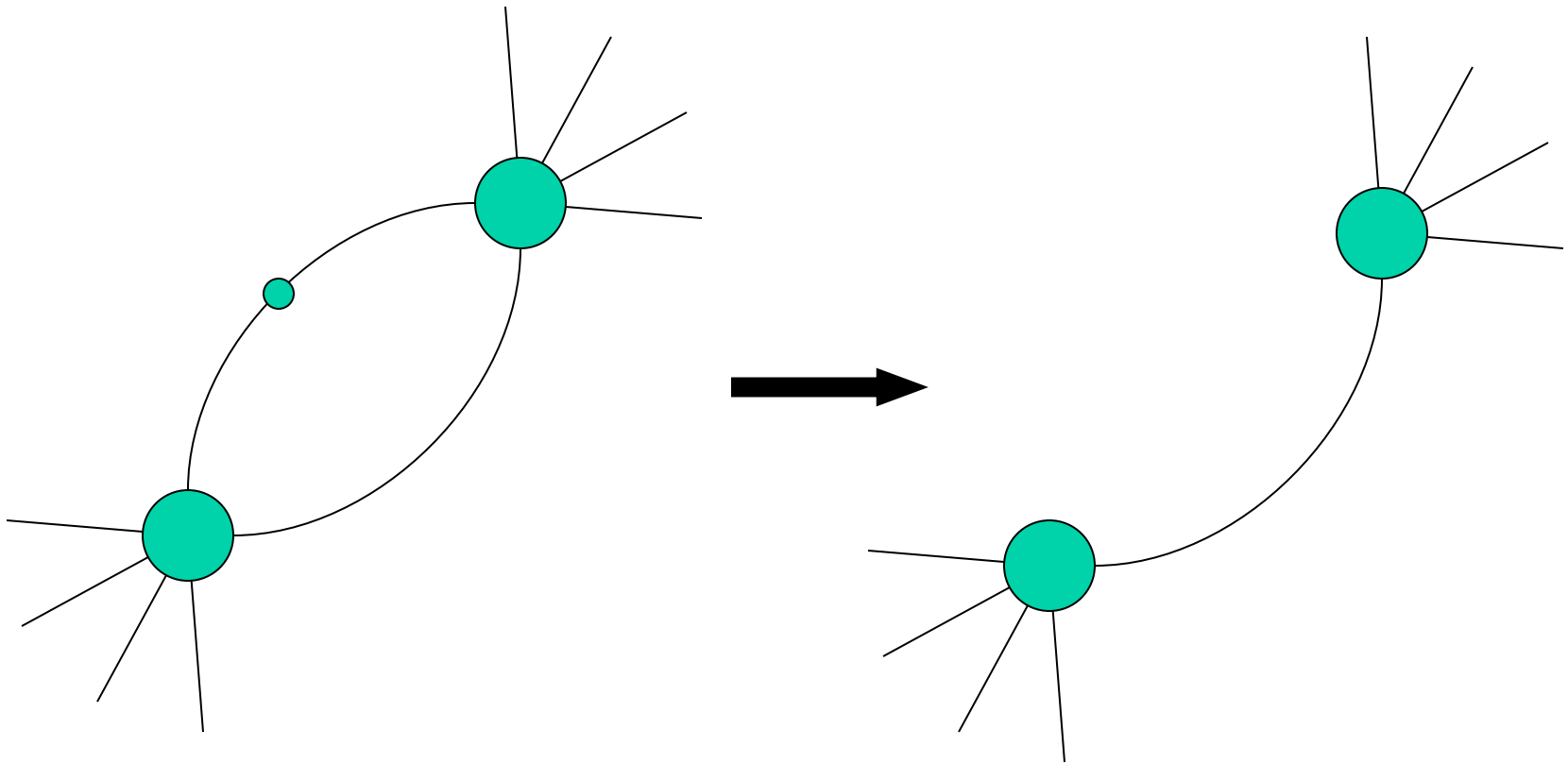
# Degree 1 Vertices



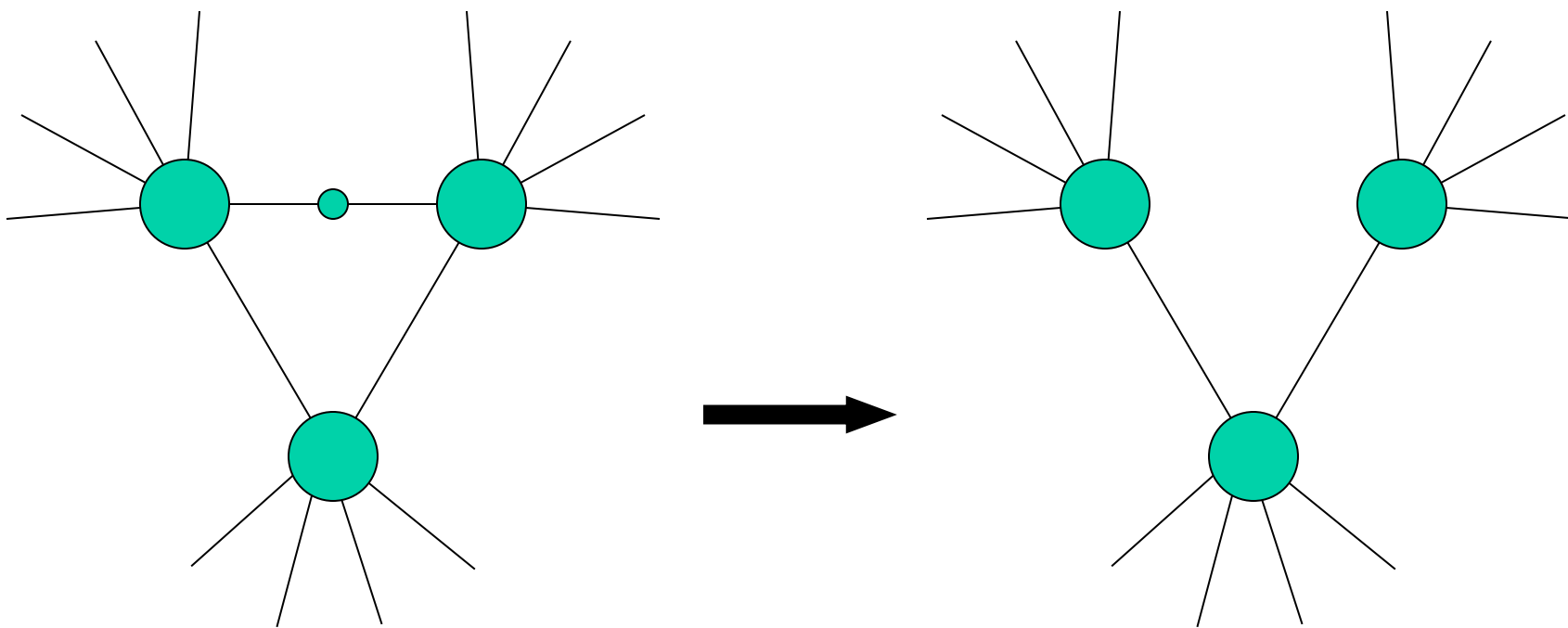
# Degree 2 Vertices



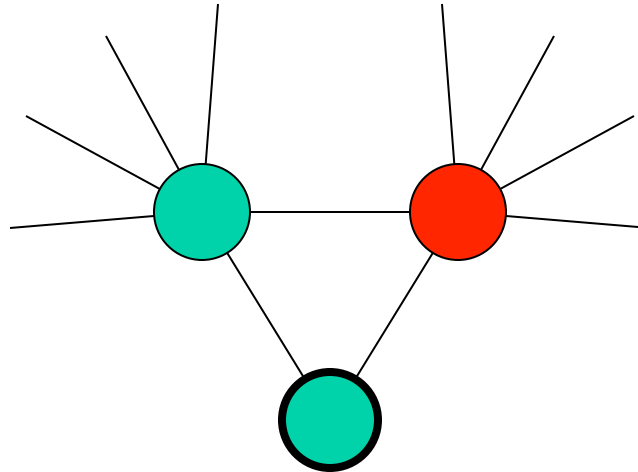
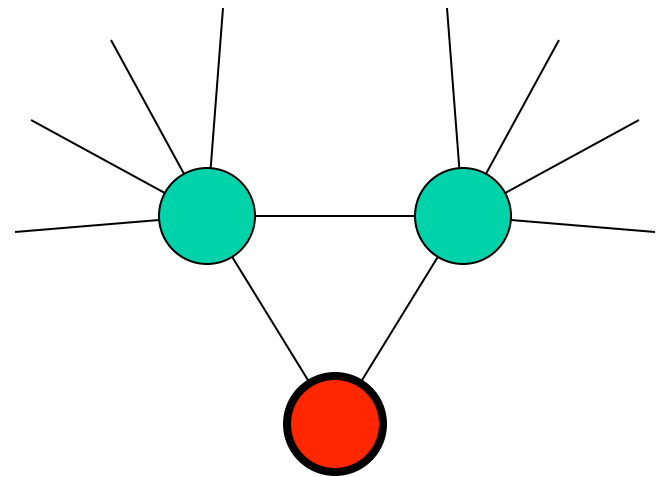
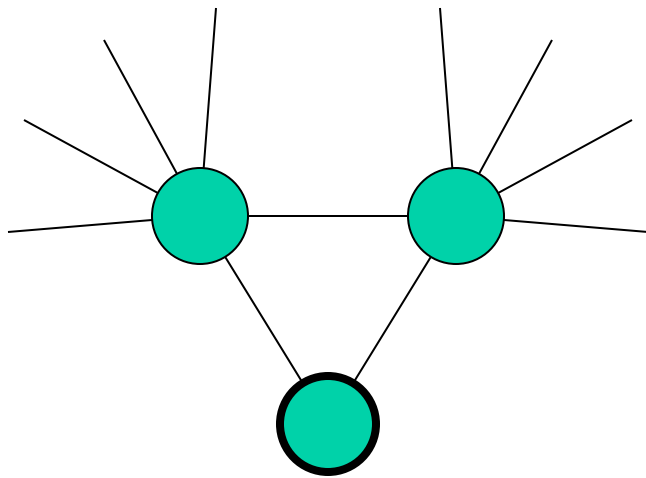
# Double Edges



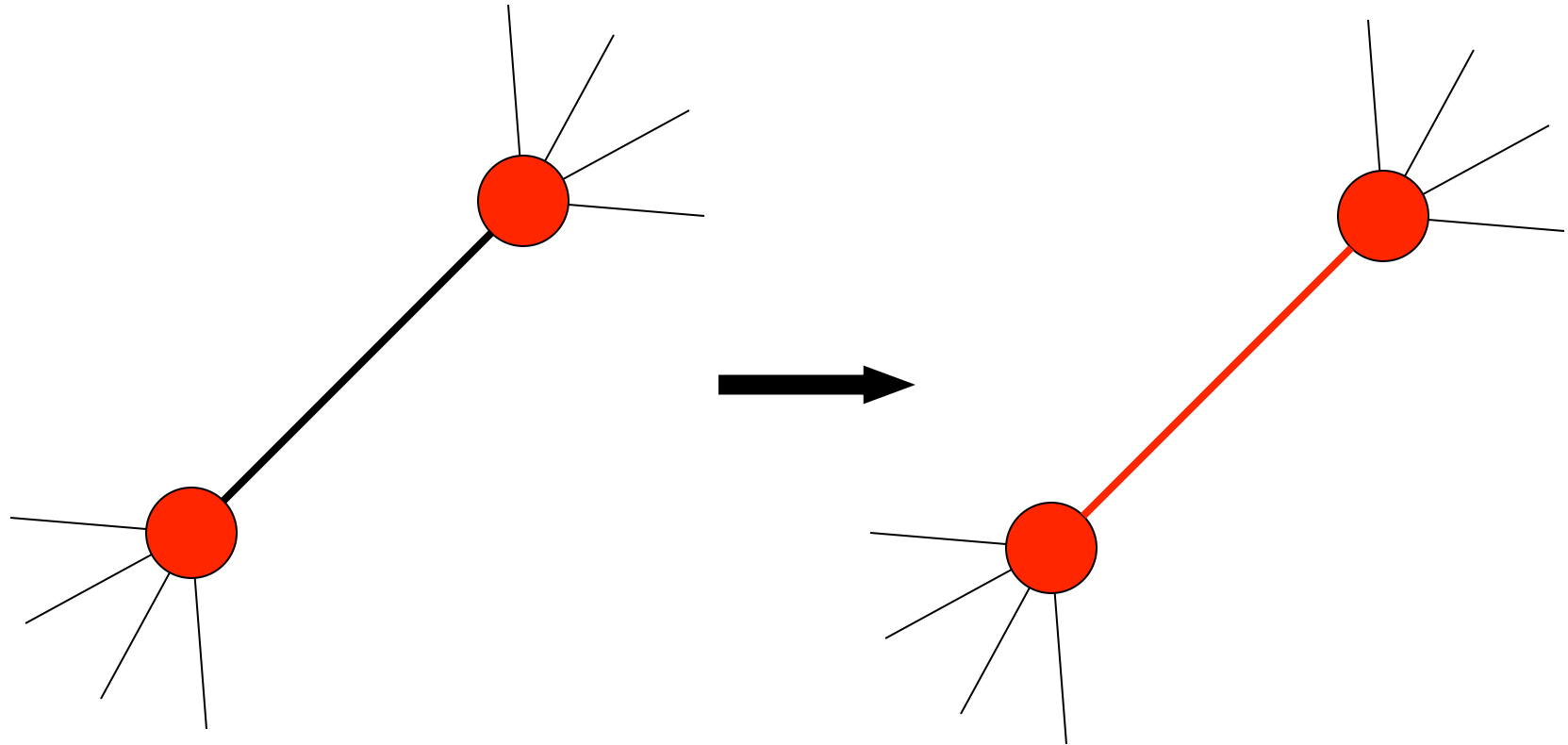
# Triangles



# Degree 2 Vertices $\Rightarrow$ Triangles

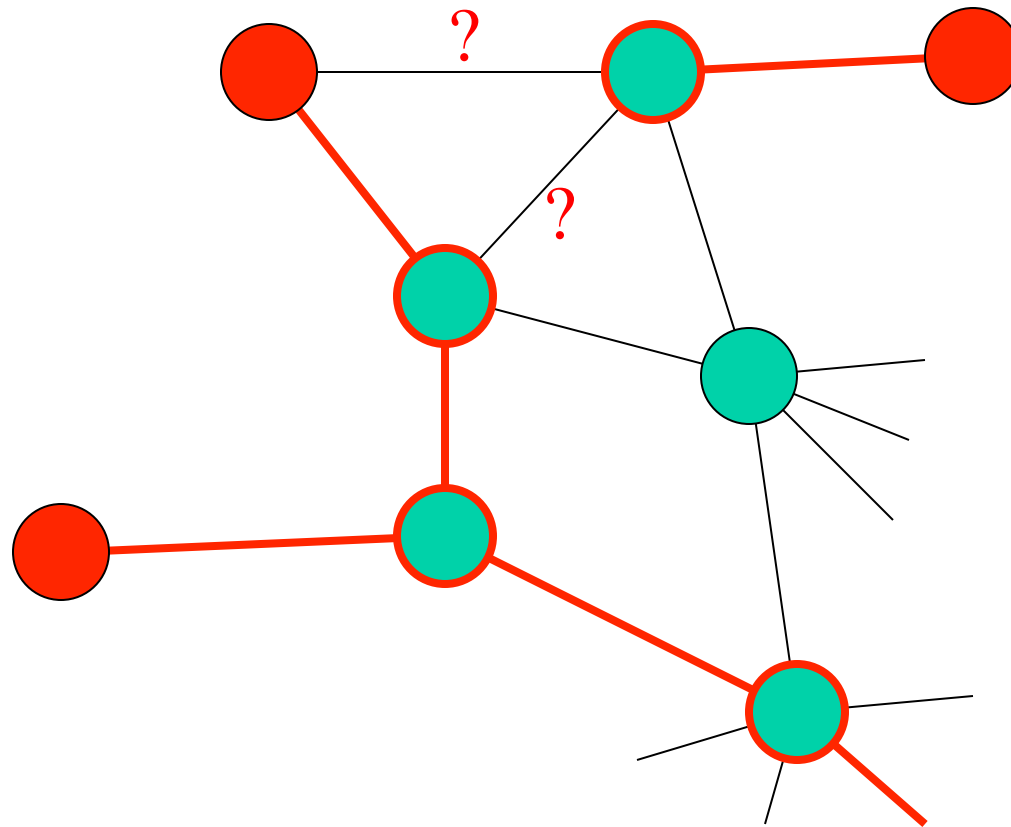


# Attaching New Vertices



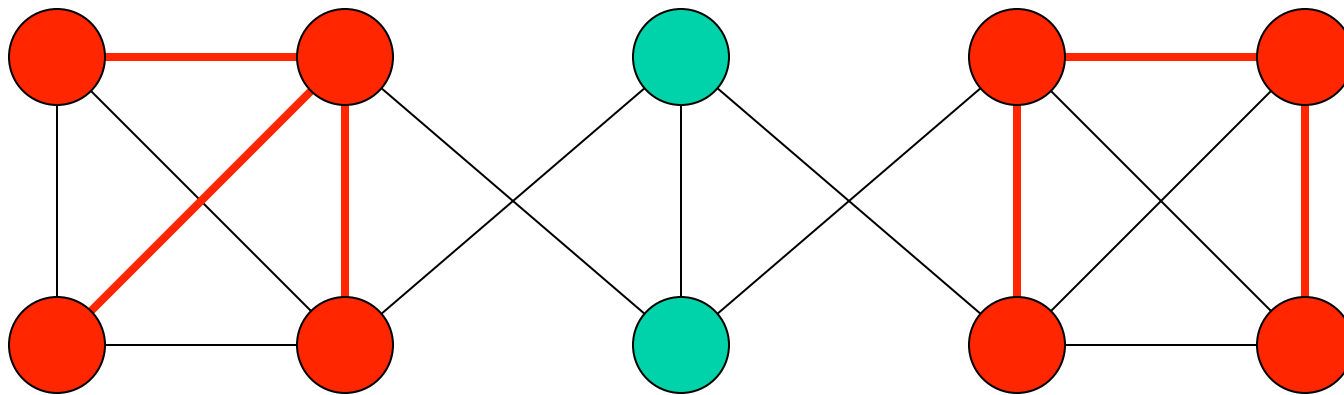
If two distinct tree-components can be connected by a single edge, it is always worth including that edge.

# Choosing an edge for Connection



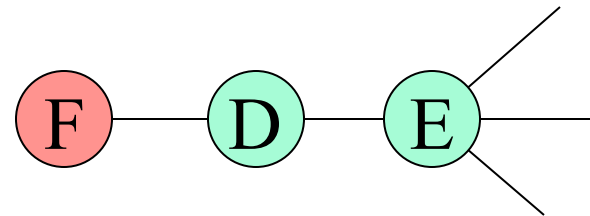
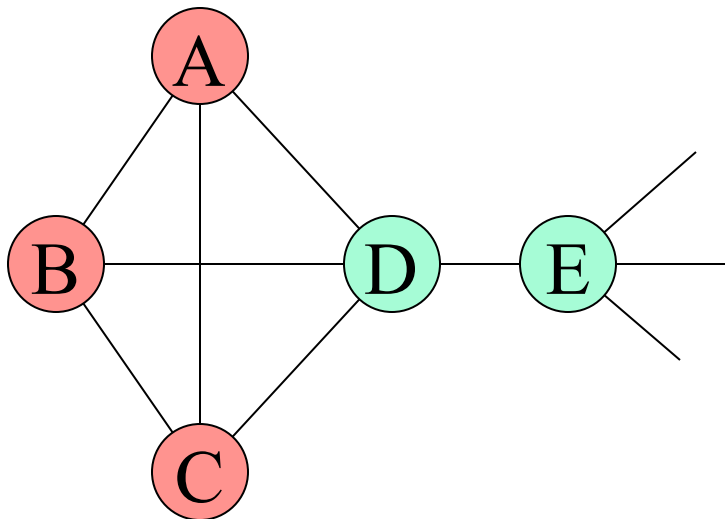
It doesn't matter *where* we connect two trees together.

# Dealing with *partial* solution trees



Remember: It doesn't matter where a vertex is connected into the tree.

Solution: Consider a tree as a single vertex!



Where **F** includes

**A** , **B** and **C**

# Tree-collapse helps!

