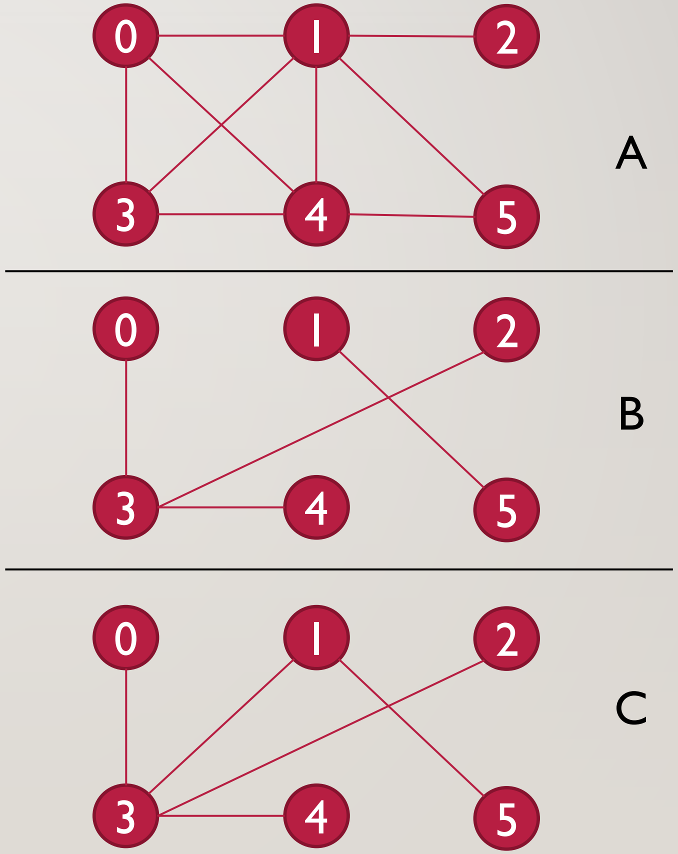
ANALYSIS OF ALGORITHMS

LECTURE 7 : TREES

BASED ON SECTION 5.1

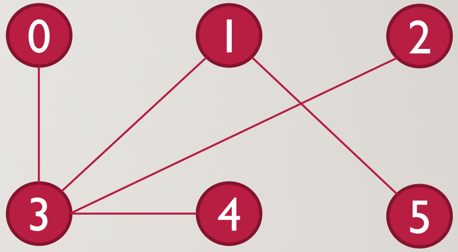
TREES

- Connected, Acyclic Graphs
- Not necessarily a binary tree
- Doesn't necessarily look like a tree

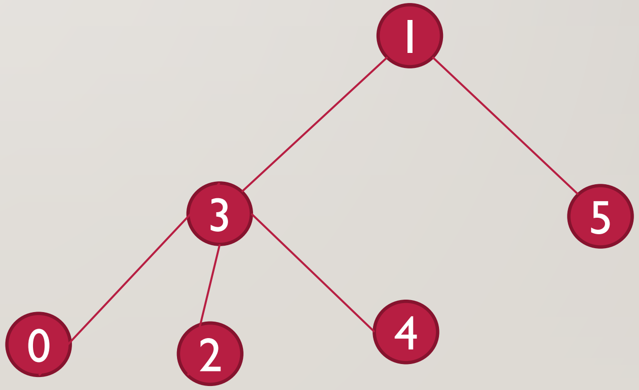


TREES

- Some vertex can be the root
- Let's say vertex 1 is the root, we could redraw the tree in A as shown in B
- There is exactly one path from a vertex to the root
- This leads to a more compact representation that we can use for trees.



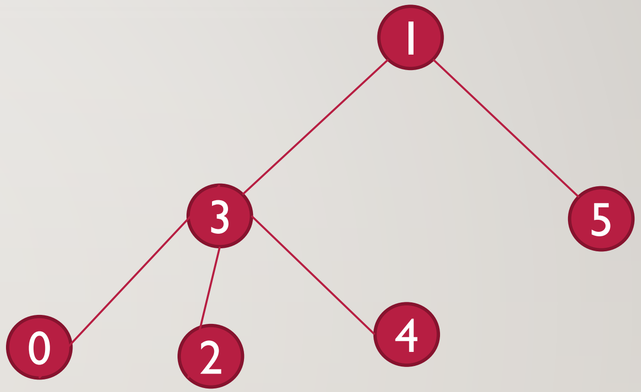
A



B

PARENT ARRAY

- Instead of using an adjacency matrix or adjacency list, we can use a parent array
- For every vertex, we keep only the first vertex on the path between it and the root (besides itself)



0	1	2	3	4	5
3	1	3	1	3	1

PROPERTIES OF TREES

- **Theorem 5.1.** Let T be a graph with n vertices. Then the following statements are equivalent.
    1. T is a tree
    2. T is connected, and has $n-1$ edges
    3. T contains no cycles and has $n-1$ edges
    4. T is connected, and every edge is a bridge
    5. Any two vertices of T are connected by exactly one path
    6. T contains no circuits, but the addition of any new edge creates exactly one circuit

OUR OLD FRIEND INDUCTION

0

- Prove 1 $\Rightarrow$ 2
- T is any graph with n vertices
- Prove T is a tree $\Rightarrow$ T is connected, and has $n-1$ edges
- Induction!
- Consider a Base Case of n=1. Now we need to show that all trees with 1 vertex have 0 edges
- This is kind of trivially true. If there were any edges, we'd have a cycle as the only possible edge is from 0 to itself. But we know that T is a tree, so it can't have any cycle.

INDUCTION STEP

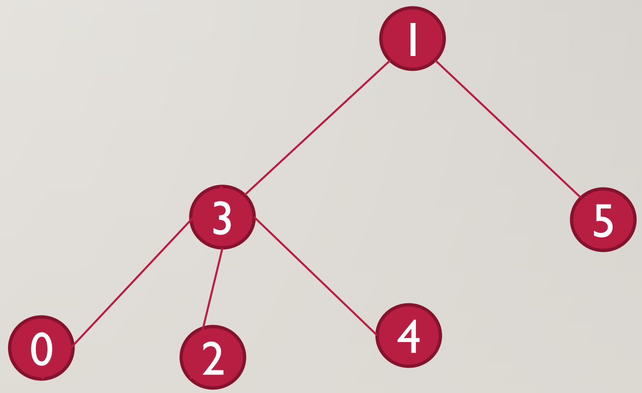
- Now we can assume 1 $\Rightarrow$ 2 for all trees with k vertices, and we need to prove that 1 $\Rightarrow$ 2 for all trees with k+1 vertices
- So we can assume all trees with k vertices have k-1 edges, and we need to prove that all trees with k+1 vertices have k edges
- We will use a similar strategy to what we used for the graph colouring proof in the previous lecture
- **Let T be a tree with k+1 vertices**

INDUCTION STEP

- Let T be a tree with k+1 vertices
- We want to prove that it has k edges, but we know nothing about trees with k+1 vertices
- We do know stuff about trees with k vertices, so we want to do something to the tree to turn it into a tree with k vertices
- Remove a vertex v from T, to produce T'
- Now, T' has k vertices. But is T' necessarily a tree?
  - Is it necessarily acyclic?
  - Is it necessarily connected?

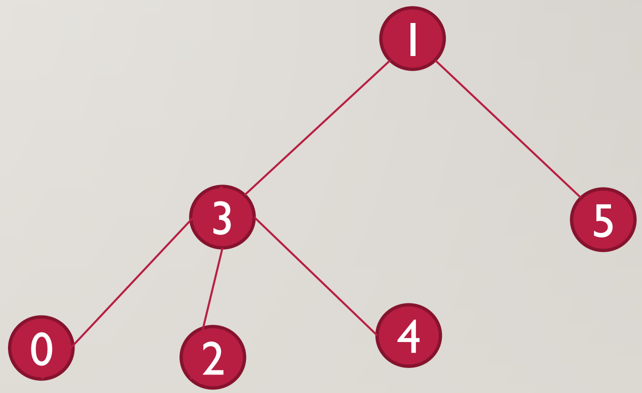
INDUCTION STEP

- Let T be a tree with k+1 vertices
- We want to prove that it has k edges, but we know nothing about trees with k+1 vertices
- We do know stuff about trees with k vertices, so we want to do something to the tree to turn it into a tree with k vertices
- Remove a vertex v from T, to produce T'
- Now, T' has k vertices. But is T' necessarily a tree?
  - Is it necessarily acyclic?
  - **Is it necessarily connected? NO!**



INDUCTION STEP

- The key is not to remove a random vertex v. Instead remove a leaf node v
- Any leaf node will have only 1 edge, so we will remove that 1 edge as well
- So when we remove v from T to produce T', T' will have k vertices
  - Is T' acyclic?
  - Is T' connected?
- Now, if T' is a tree with k vertices, we know it has k-1 edges
- But T has exactly one more edge than T', so it has k edges
- So every tree with k+1 vertices has k edges
- So our induction meme has gone viral
- So 1 ⇒ 2 for all trees of size k, for all natural numbers k
- So 1 ⇒ 2 for all trees



READING

- **Read Sections 5.1 (Properties of Trees) and 5.3 (Search Trees) in the book**