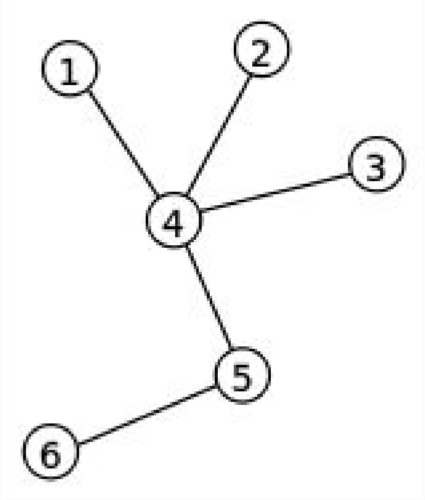
Trees

Problem Solving Club
November 30, 2016

What is a tree?

A tree is an **undirected** graph. The following are all equivalent definitions:

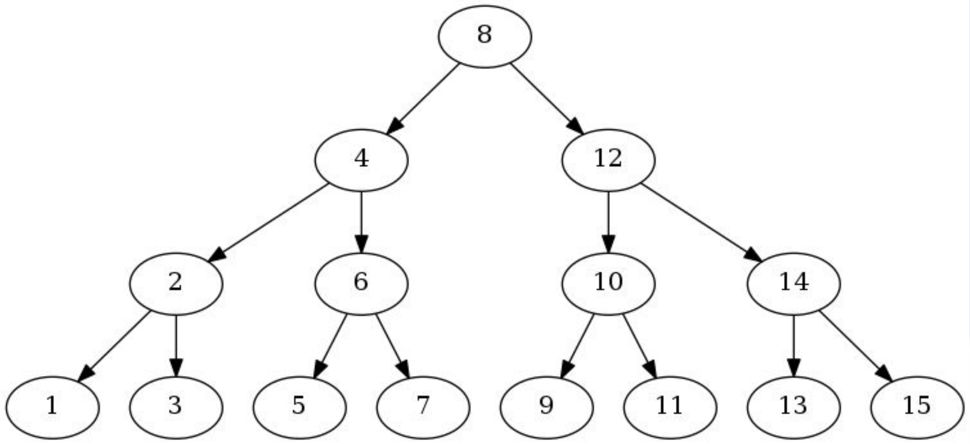
- Any two vertices are connected by exactly one path
- Connected with exactly V-1 edges
- Connected and has no cycles



Which is better: adjacency list or matrix?

- Adjacency list: O(V+E) = O(V) space
- Adjacency matrix: O(V^2) space

Basic tree definitions

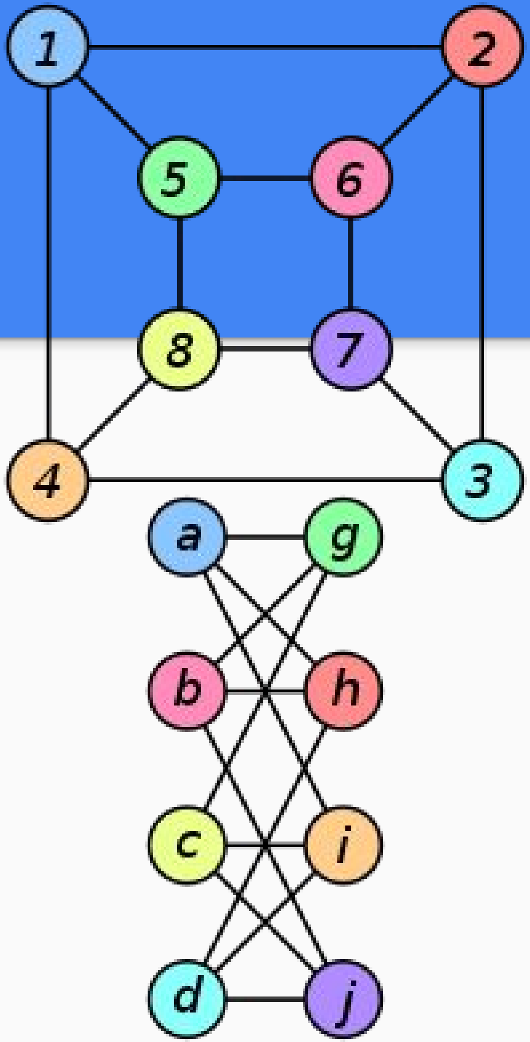


- A **rooted tree** has a **root** which can be any vertex
- A **leaf** is a vertex of degree 1
- The **height** of a rooted tree is the length of the longest downward path from the root to a leaf from that vertex
- The **depth** of a vertex is the length of the path to the root

Why are trees special?

- Many problems on general graphs are easy on trees
  - Graph isomorphism: Basically, whether two graphs are the "same" but labelled differently
  - Graph coloring: Find the minimum number of colors needed to color a graph so that two adjacent vertices have different colors

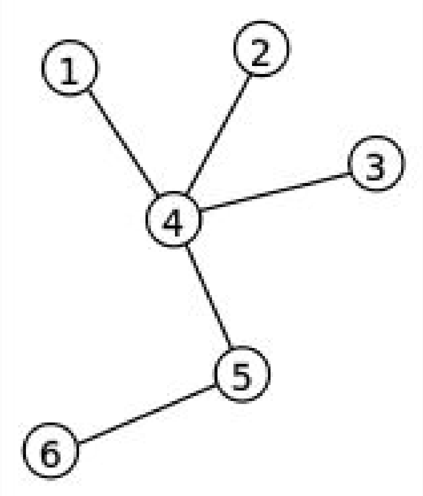
    What is the minimum number of colors needed for a tree?
- Trees are commonly asked in interviews



Diameter of a tree

Given a tree, find the **maximum distance** between two vertices.

1. Choosing any vertex (v)
2. Find any furthest vertex from v (v_1)
3. Find any furthest vertex from v_1 (v_2)
4. The tree diameter is the distance from v_1 to v_2

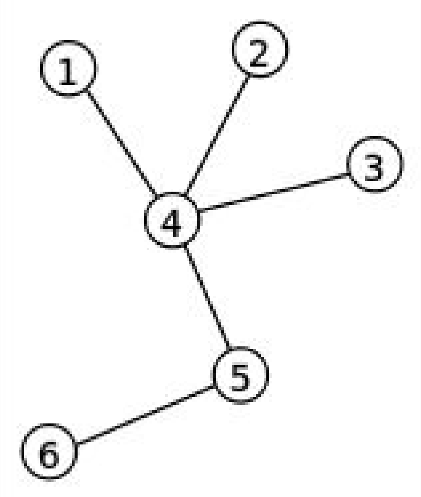


Radius of a tree

Given a tree, find a **best root** (center), which is a root that minimizes the height of the tree.

How many best roots can there be?

1. Find the tree diameter
2. The midpoint(s) of the path are the best root(s)

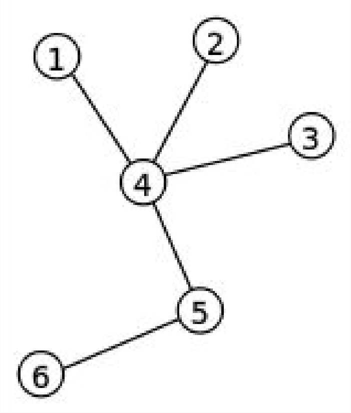


Eccentricity

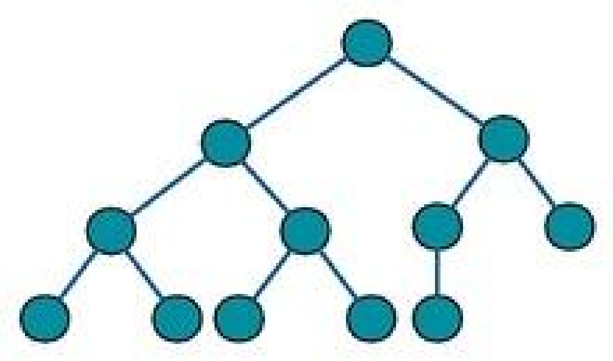
The **eccentricity** of the vertex v is the maximum distance from v to any vertex

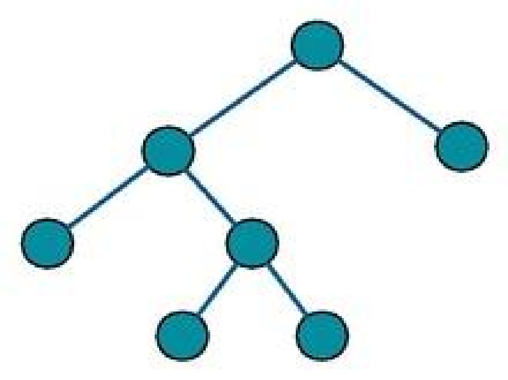
- So, the **diameter** is the maximum eccentricity among all vertices
- The **radius** is the minimum eccentricity among all vertices

The eccentricity of a tree can be found in O(V) time



Binary tree





A **binary tree** is a tree in which every vertex has at most two children

- Commonly used to implement data structures, like:
  Binary search tree, binary heap, segment tree
- A **full** binary tree is where every vertex has 0 or 2 children
- A **complete** binary tree is where every level is filled (except maybe the last)

Symmetric tree

Given a binary tree, check whether it is a mirror of itself

For example, symmetric:

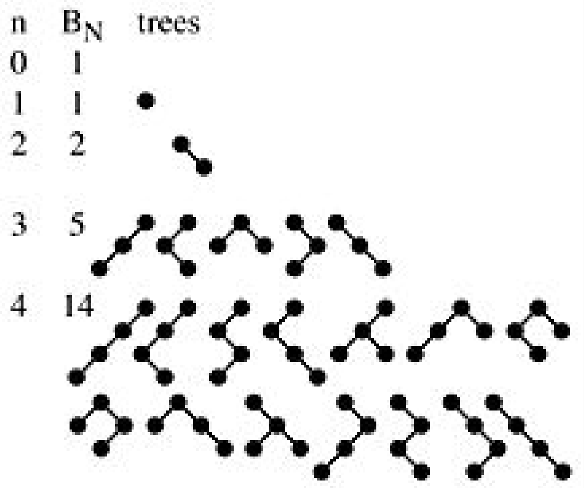
```
    1
   / \
  2   2
 / \ / \
3  4 4  3
```

But the following is not:

```
  1
 / \
2   2
 \   \
 3    3
```

Unique binary trees

Given n, how many binary trees are there with *n* nodes?



n	B_N	trees
0	1	
1	1	
2	2	
3	5	
4	14	