Graph Theory

Trees

Shengfa Wang
sfwang@dlut.edu.cn

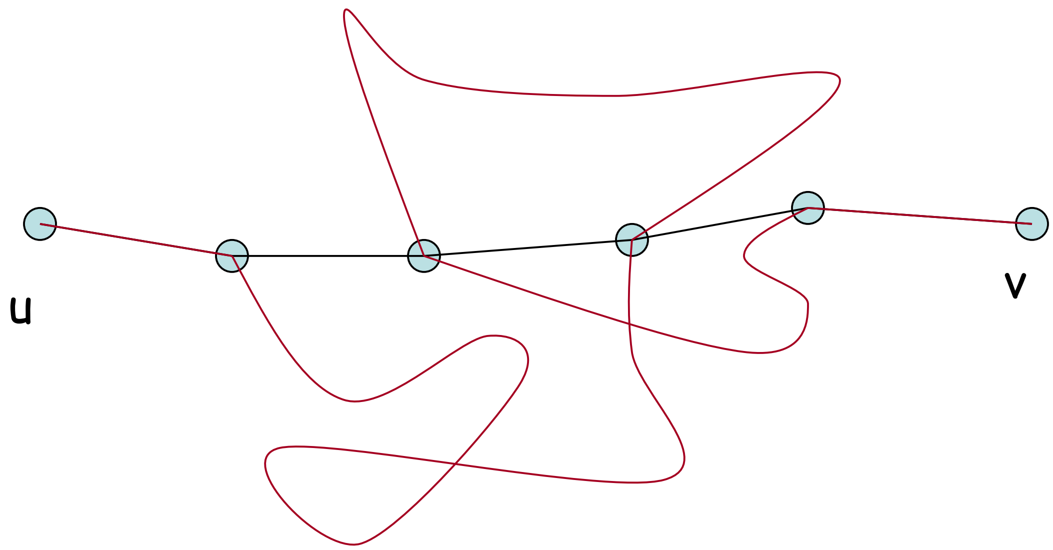
Tree Characterization by Path

树是没有循环的连通图

Definition. A tree is a connected graph with no cycles.

Can there be no path between u and v? NO

Can there be more than one simple path between u and v? NO



This will create cycles.

一条路径上的节点、除了起点和终点可以相同以外其他节点均不相同

起点和终点相同的简单路径称为简单回路.

在树中，每对顶点之间都有一条唯一的简单路径

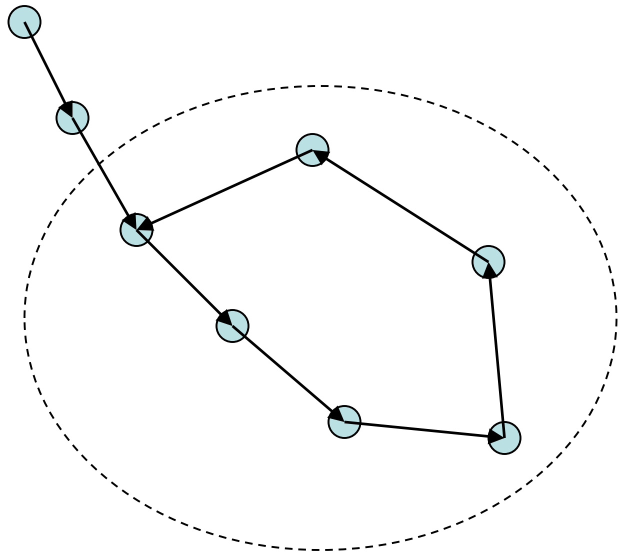
Claim. In a tree, there is a unique simple path between every pair of vertices.

Tree Characterization by Number of Edges

Definition. A tree is a connected graph with no cycles.

Can a tree have no leaves? NO

Then every vertex has degree at least 2.



Go to unvisited edges as long as possible.

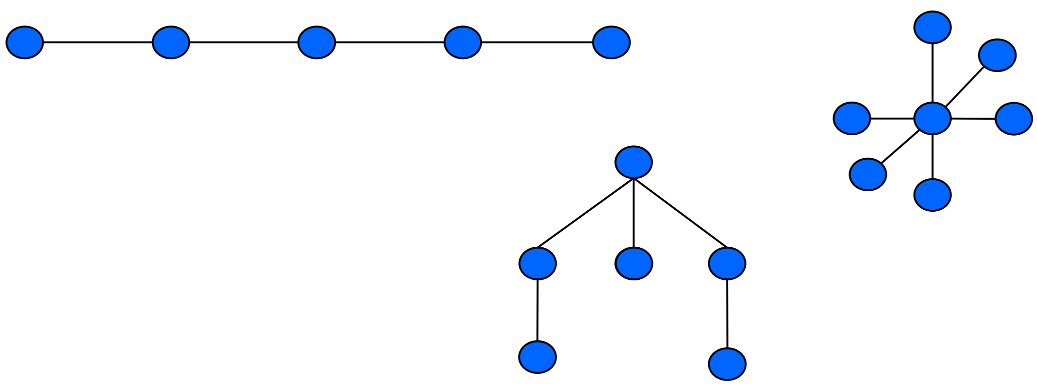
Cannot get stuck,
unless there is a cycle.

Tree Characterization by Number of Edges

Definition. A tree is a connected graph with no cycles.

Can a tree have no leaves? NO

How many edges does a tree have? n-1?



我们通常使用n表示顶点数，并使用m表示图中的边数。

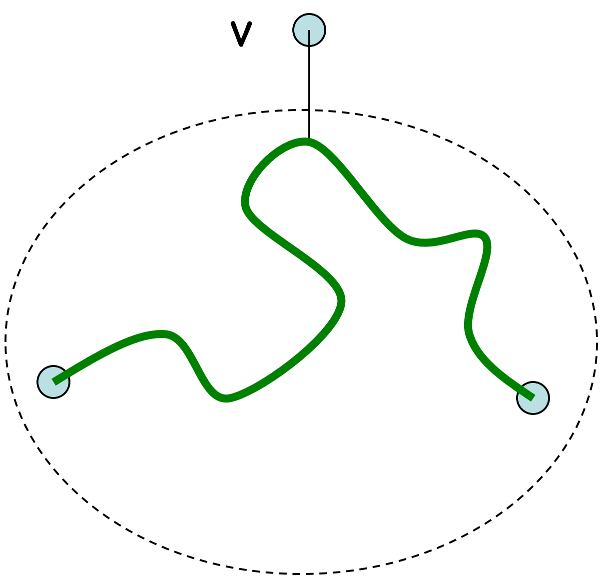
We usually use n to denote the number of vertices, and use m to denote the number of edges in a graph.

Tree Characterization by Number of Edges

Definition. A tree is a connected graph with no cycles.

Can a tree have no leaves? NO

How many edges does a tree have? n-1?



Look at a leaf v.

Is T-v a tree? YES

1. Can T-v has a cycle? NO
2. Is T-v connected? YES

By induction, T-v has (n-1)-1=n-2 edges.

So T has n-1 edges.

Tree Characterizations

Definition. A tree is a connected graph with no cycles.

Characterization by paths:

A graph is a tree if and only if

①任意两个点之间有且仅有一条 simple path
则其一定是树

there is a unique simple path between every pair of vertices.

Characterization by number of edges:

② A graph is a tree if and only if it is connected and has n-1 edges.

(We have only proved one direction.

The other direction is similar and left as an exercise.)

① 树是 没有循环的连通图、无向图

Trees

•**Definition:** A **tree** is a connected undirected graph with no simple circuits.

一棵树不能包含多个边或循环

•Since a tree cannot have a simple circuit, a tree cannot contain multiple edges or loops.

•Therefore, any tree must be a **simple graph**. 任何树都是简单图

无向的
没有自循环
没有多重边

•**Theorem:** An undirected graph is a tree if and only if there is a **unique simple path** between any of its vertices.

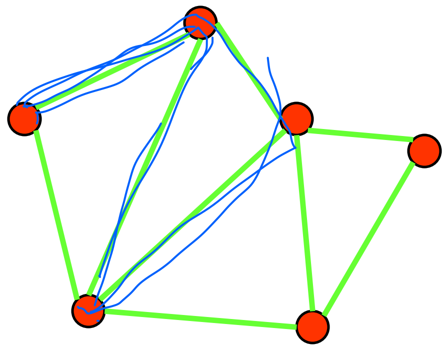
•**Definition:** An undirected graph that does not contain simple circuits and is not necessarily connected is called a **forest**.

通常，我们使用树来表示层次结构

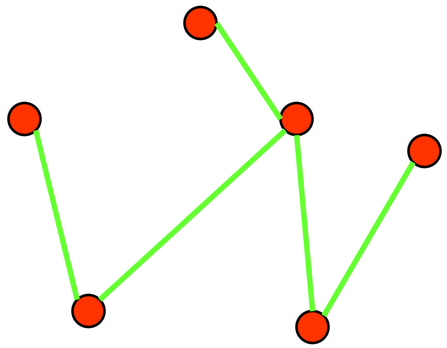
•In general, we use trees to represent **hierarchical structures**.

Trees

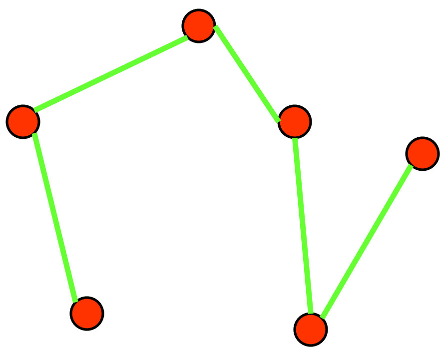
•**Example:** Are the following graphs trees?



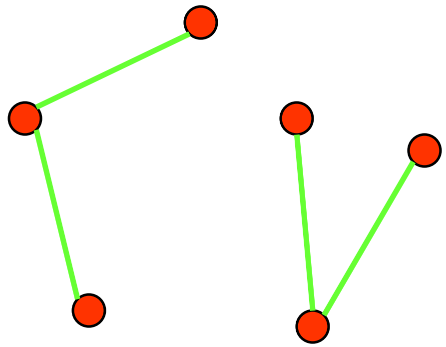
No.



Yes.



Yes.



No.

生成树

Spanning Trees

令G为简单图形。 G的生成树是G的子图，G是包含G的每个顶点的树。

•**Definition:** Let G be a simple graph. A spanning tree of G is a subgraph of G that is a tree containing every vertex of G.

G =（V，E）的生成树是V上具有最小边数的连接图

•**Note:** A spanning tree of G = (V, E) is a connected graph on V with a minimum number of edges
(|V| - 1).

•**Example:** Since winters in Boston can be very cold, six universities in the Boston area decide to build a tunnel system that connects their libraries.

无向图的点不动，去掉边，直到边的个数为n-1为止

Spanning tree

无向图

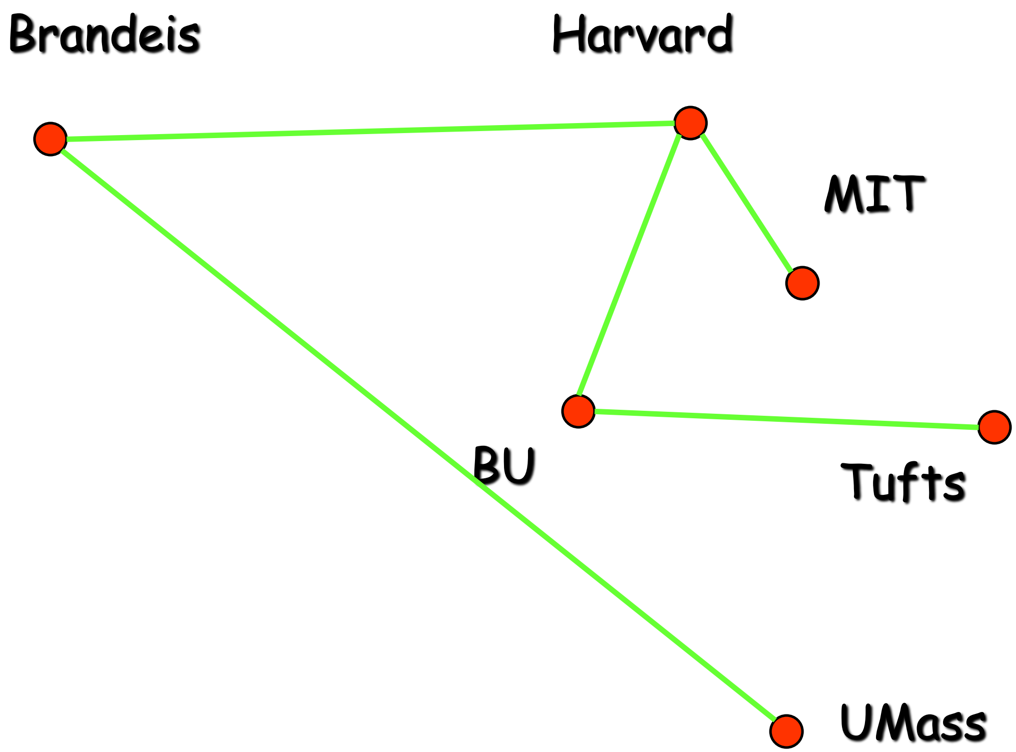
- A spanning tree in an undirected graph G(V,E) is a subset of edges $T \subseteq E$ that are acyclic and connect all the vertices in V.
- A spanning tree must consist of exactly n-1 edges.
- Suppose that each edge has a 权重 weight associated with it. Say that the weight of a tree T is the sum of the weights of its edges $w(T) = \sum_{e \in T} w(e)$

最小生成树

- The minimum spanning tree in a weighted graph G(V,E) is one which has the smallest weight among all spanning trees in G(V,E)

Spanning Trees

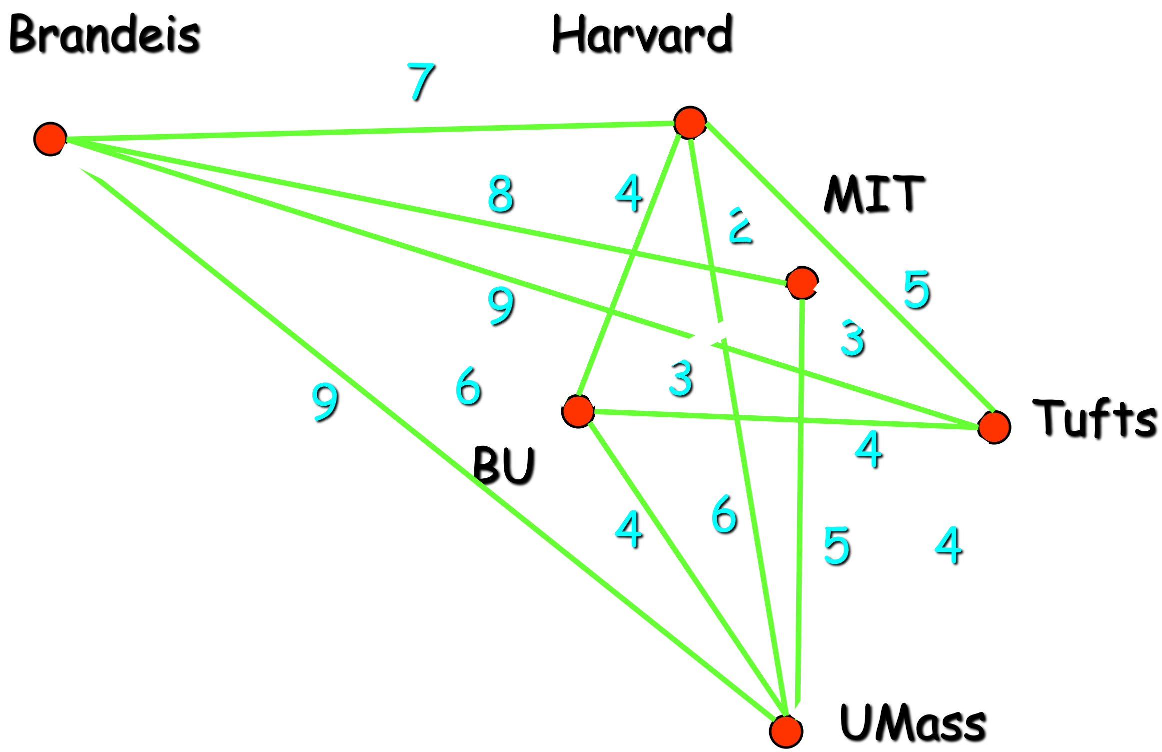
•Example for a spanning tree:



Since there are 6 libraries, 5 tunnels are sufficient to connect all of them.

Spanning Trees

- The complete graph with cost labels (in billion $):



The least expensive tunnel system costs $20 billion.

Spanning Trees

- Now imagine that you are in charge of the tunnel project. How can you determine a tunnel system of **minimal cost** that connects all libraries?
- **Definition:** A **minimum spanning tree** in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges.
- How can we find a minimum spanning tree?

Spanning Trees

①先把所有边排序
②选最小的边

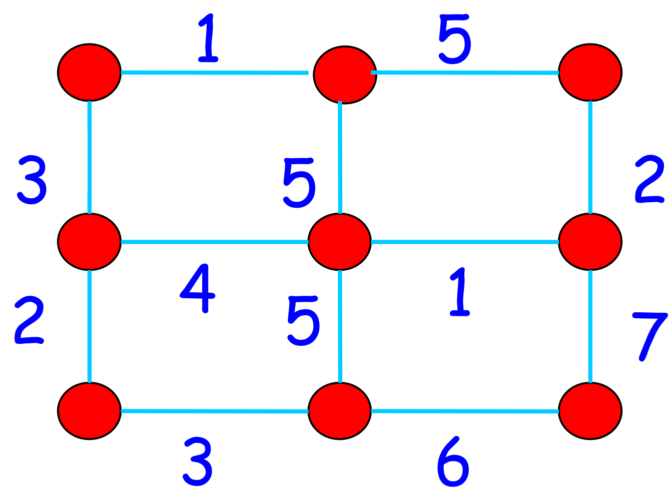
•Prim's Algorithm: ③选与这条边相邻的权重小的边

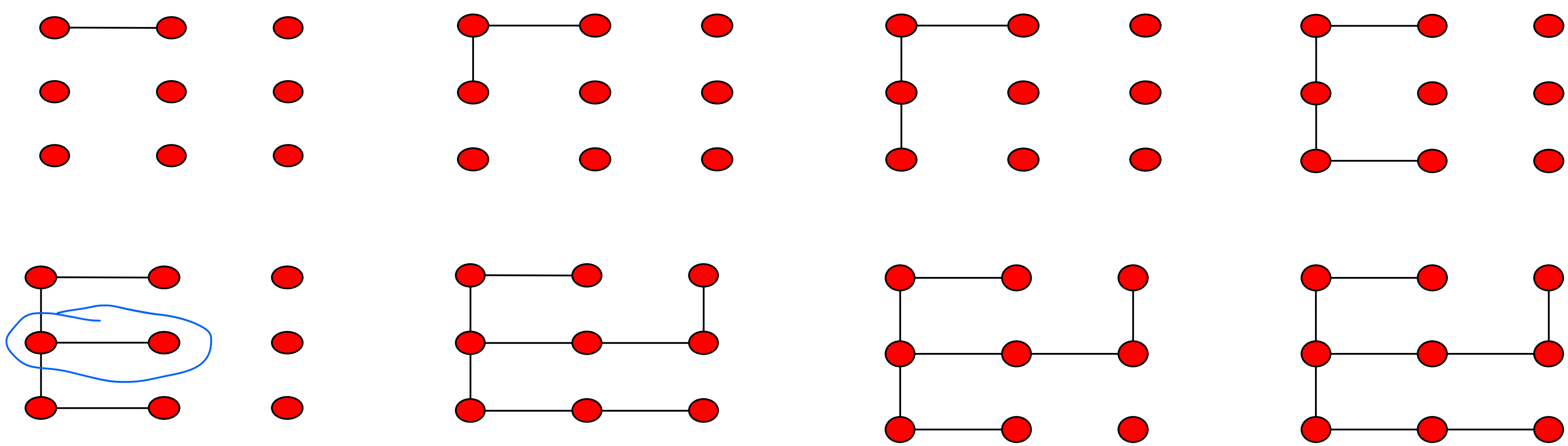
• Begin by choosing any edge with **smallest weight** and putting it into the spanning tree,

• successively add to the tree edges of **minimum weight** that are incident to a vertex already in the tree and not forming a simple circuit with those edges already in the tree,

• stop when (n – 1) edges have been added.

Prim's algorithm





Spanning Trees

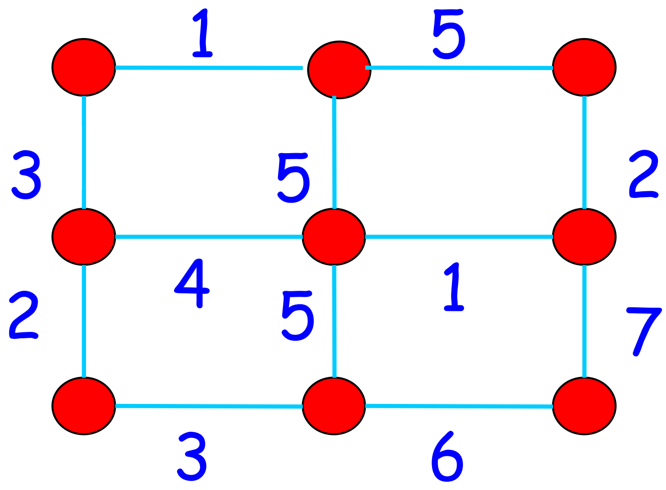
•Kruskal's Algorithm:

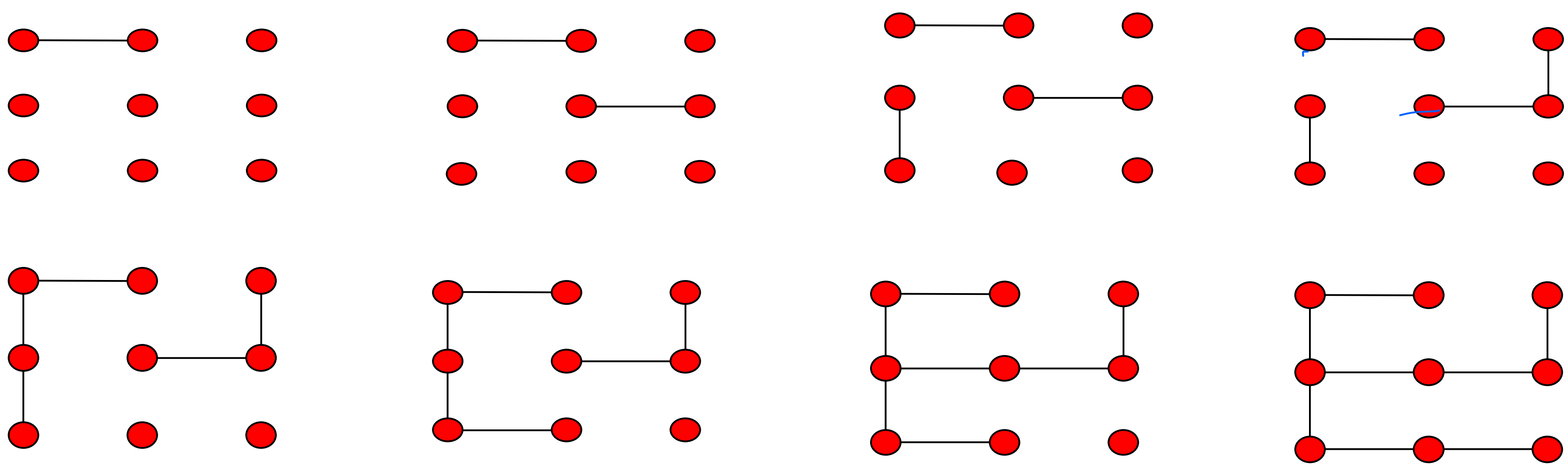
•Kruskal's algorithm is identical to Prim's algorithm, except that it does not demand new edges to be incident to a vertex already in the tree.

•Both algorithms are **guaranteed** to produce a minimum spanning tree of a connected weighted graph.

先把各边权重从小到大排列。然后从权重最小的边开始加

Kruskal's algorithm





Rooted Trees 根树（有向树）

- We often designate a particular vertex of a tree as the **root**. Since there is a unique path from the root to each vertex of the graph, we direct each edge away from the root.
- Thus, a tree together with its root produces a **directed graph** called a **rooted tree**.

根：入度为0

其他入度均为1

叶：出度为0

分枝点/内点：出度不为0

Rooted Trees

- If v is a vertex in a rooted tree other than the root, the **parent** of v is the unique vertex u such that there is a directed edge from u to v.
- When u is the parent of v, v is called the **child** of u.
- Vertices with the same parent are called **siblings**.
- The **ancestors** of a vertex other than the root are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root.

Rooted Trees

- The **descendants** of a vertex v are those vertices that have v as an ancestor.
- A vertex of a tree is called a **leaf** if it has no children.
- Vertices that have children are called **internal vertices**.
- If a is a vertex in a tree, then the **subtree** with a as its root is the subgraph of the tree consisting of a and its descendants and all edges incident to these descendants.

如果a是树中的顶点，则以a为根的子树是树的子图，该子图由a及其后代以及入射到这些后代的所有边组成。

Rooted Trees

level: •根树中顶点v~~的级别~~是从根到此顶点的唯一路径的长度。

- The **level** of a vertex v in a rooted tree is the length of the unique path from the root to this vertex.

- The level of the root is defined to be zero.

高是最大的level

- The **height** of a rooted tree is the maximum of the levels of vertices.

节点深度：从根节点到该节点最长简单路径边的条数

高度：从某点到叶节点的最长简单路径边的条数

深度为n，最多有 2^n-1 个节点（根节点深度按1算）

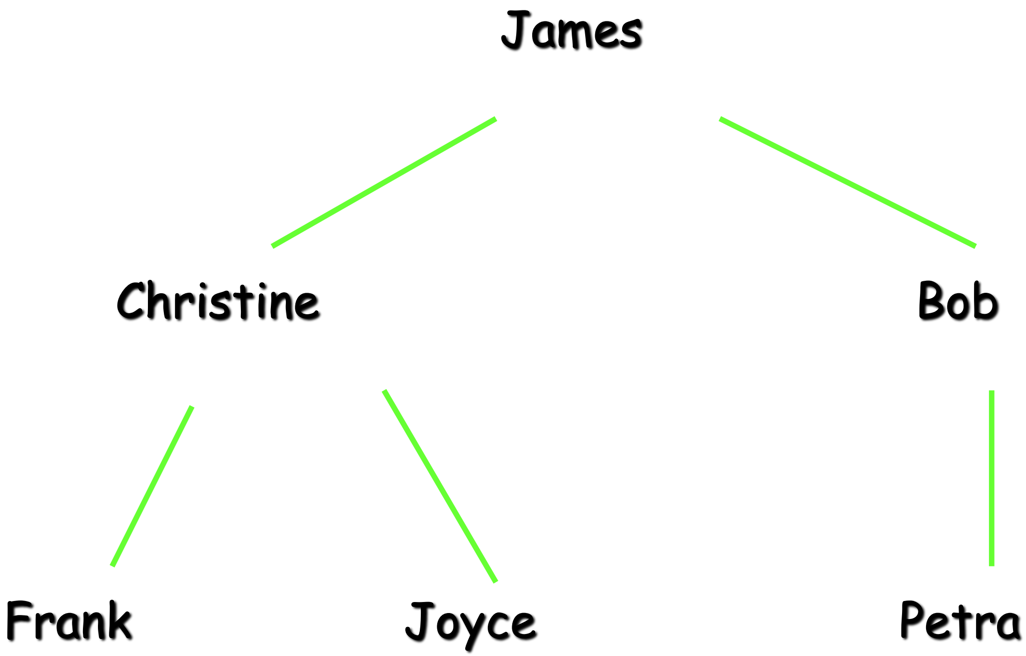
在第i层，最多有 2^{i-1} 个节点

具有n个结点的二叉树最小深度为 $\log_2(n+1)$

最大深度为 $N-1$

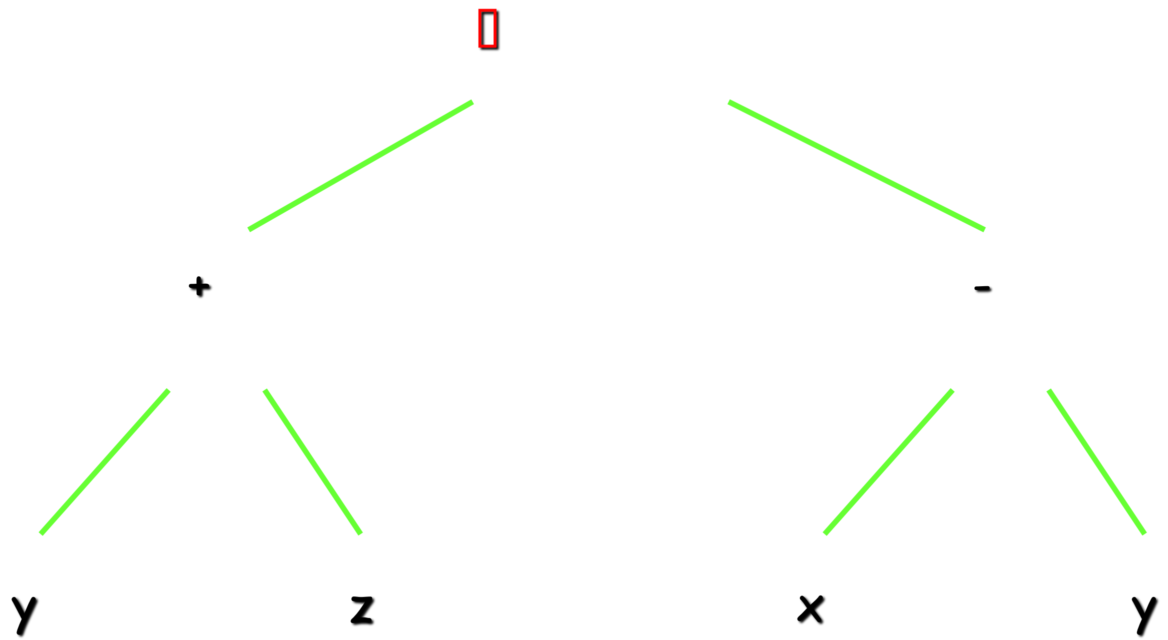
Trees

- **Example I:** Family tree



Trees

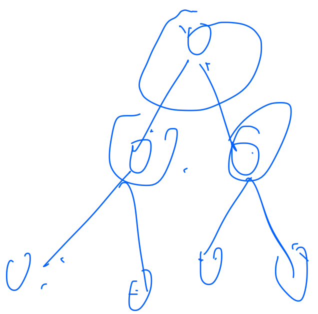
- **Example III:** Arithmetic expressions



This tree represents the expression (y + z)□(x - y).

Trees m叉树

- **Definition:** A rooted tree is called an **m-ary tree** if every internal vertex has no more than m children.
- The tree is called a **full m-ary tree** if every internal vertex has exactly m children. 每个点出度均为m
- An m-ary tree with m = 2 is called a **binary tree**.
- **Theorem:** A tree with n vertices has (n – 1) edges.
- **Theorem:** A full m-ary tree with i internal vertices contains n = mi + 1 vertices.



3x2+1

2

6

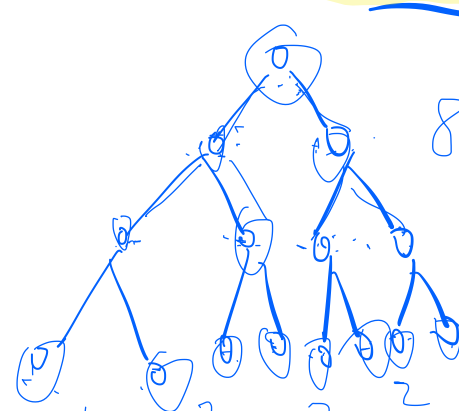
4+

13

13

6x2

12+1 13



8+

7x2

15

Binary Trees

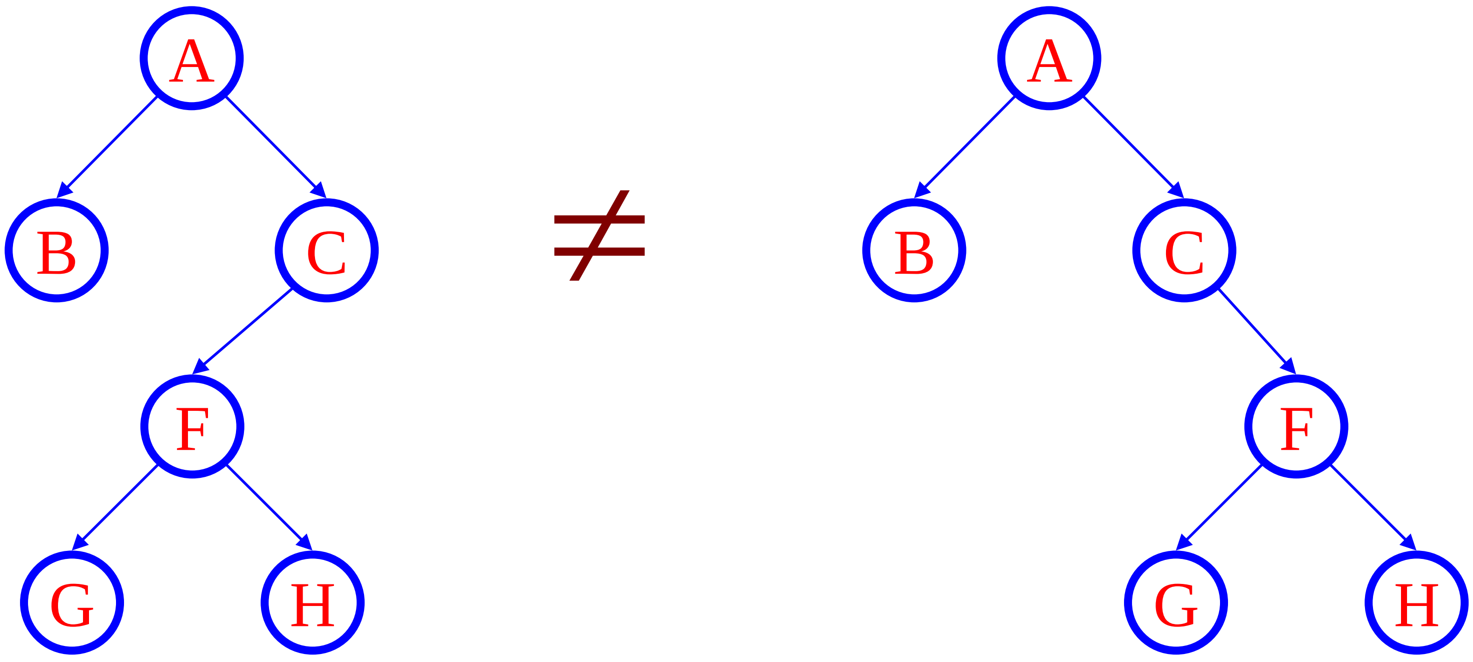
- Every node has at most two children 每个节点最多有2个孩子.
- Most popular tree in computer science
- Given N nodes, what is the minimum depth of a binary tree? $\log_2(N+1)$

最大的深度 二叉树有N个节点

- What is the maximum depth of a binary tree with N nodes? $N-1$.

Binary Trees

- Notice:
- we distinguish between left child and right child



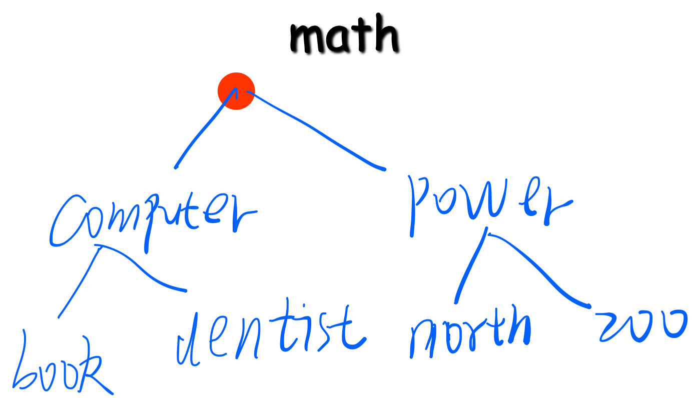
Binary Search Trees 二叉搜索树

- If we want to perform a large number of searches in a particular list of items, it can be worthwhile to arrange these items in a **binary search tree** to facilitate the subsequent searches.
- A binary search tree is a binary tree in which each child of a vertex is designated as a **right or left child**, and each vertex is labeled with a **key**, which is one of the items.
- When we construct the tree, vertices are assigned keys so that the key of a vertex is both larger than the keys of all vertices in its left subtree and smaller than the keys of all vertices in its right subtree.

小放左
大放右

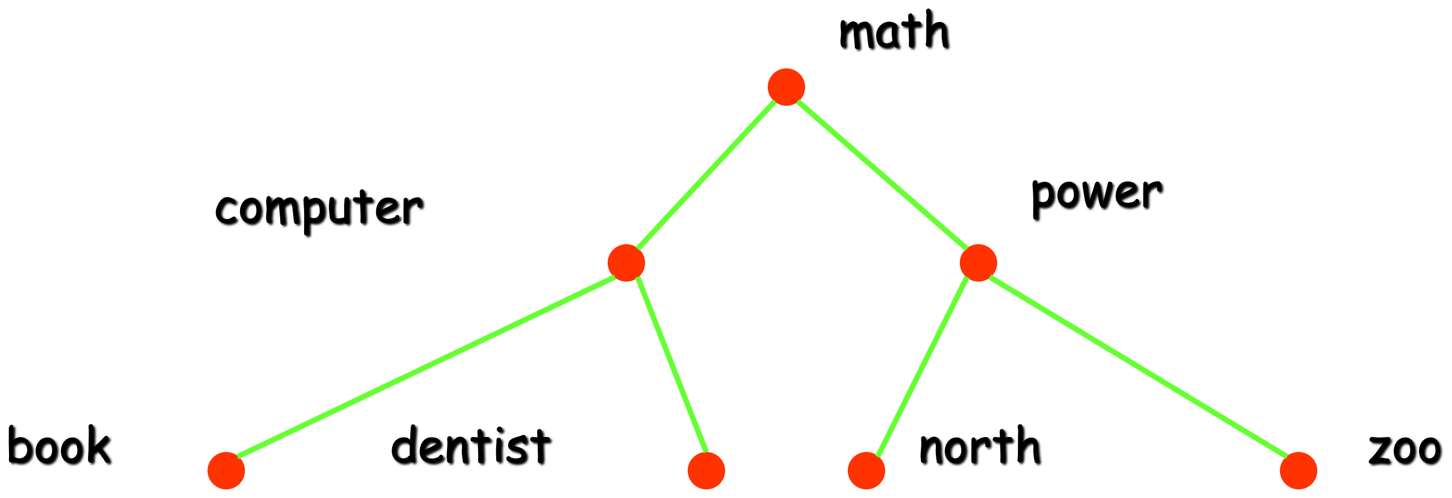
Binary Search Trees

- **Example:** Construct a binary search tree for the strings **math, computer, power, north, zoo, dentist, book**.



Binary Search Trees

•**Example:** Construct a binary search tree for the strings **math, computer, power, north, zoo, dentist, book**.



Binary Search Trees

小放左

- To perform a search in such a tree for an item x, we can start at the root and compare its key to x. If x is **less** than the key, we proceed to the **left** child of the current vertex, and if x is **greater** than the key, we proceed to the **right** one.

大放右

- This procedure is repeated until we either found the item we were looking for, or we cannot proceed any further.

The End