Data Organization:
Creating Order Out Of Chaos

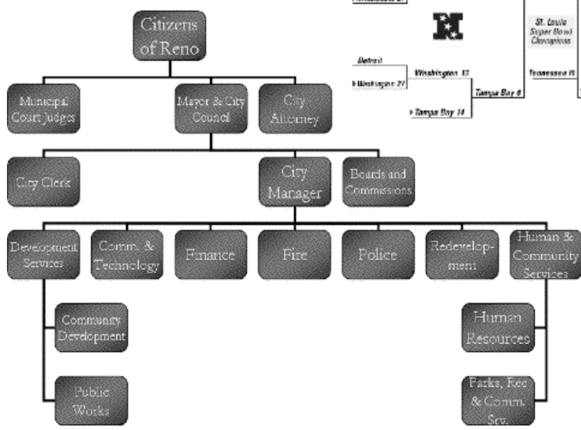
3B

Non-linear data structures

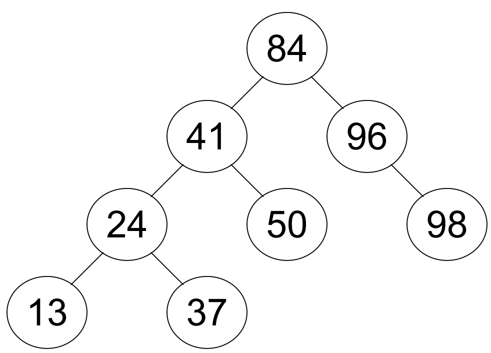
Trees

- A **tree** is a hierarchical (usually non-linear) data structure.
  - Every tree has a **node** called the **root**.
  - Each node can have 1 or more nodes as **children**.
  - A node that has no children is called a **leaf**.
- A common tree in computer science algorithms is a **binary tree**.
  - A binary tree consists of nodes that have at most 2 children.
  - A **complete binary tree** has the maximum number of nodes on each of its levels.
- Applications: data compression, file storage, game trees

Trees



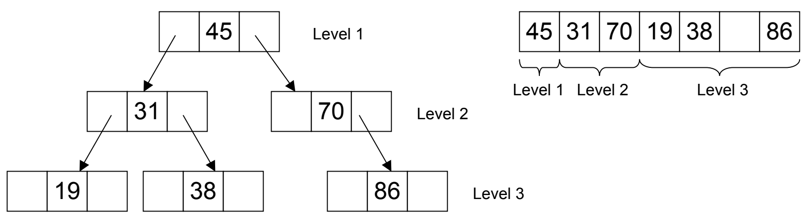
A Binary Tree



The root has the data value 84.
There are 4 leaves in this binary tree: 13, 37, 50, 98.
This binary tree is not complete.

Implementation of Trees

- One common implementation of binary trees uses nodes like a linked list does.
  - Instead of having a “next” pointer, each node has a “left” pointer and a “right” pointer.
- We could also use a vector.



Example

Binary Search Trees

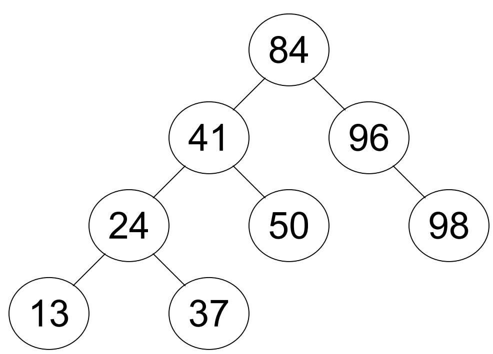
- A binary search tree (BST) is a binary tree such that
  - All nodes to the left of any node have data values less than that node
  - All nodes to the right of any node have data values greater than that node

Example
Binary Search Trees

- For each data value that you wish to insert into the binary search tree:
  - Start at the root and compare the new data value with the root.
  - If it is less, move down left. If it is greater, move down right.
  - Repeat on the child of the root until you end up in a position that has no node.
  - Insert a new node at this empty position.

Example
Binary Search Trees

- Insert: 84, 41, 96, 24, 37, 50, 13, 98

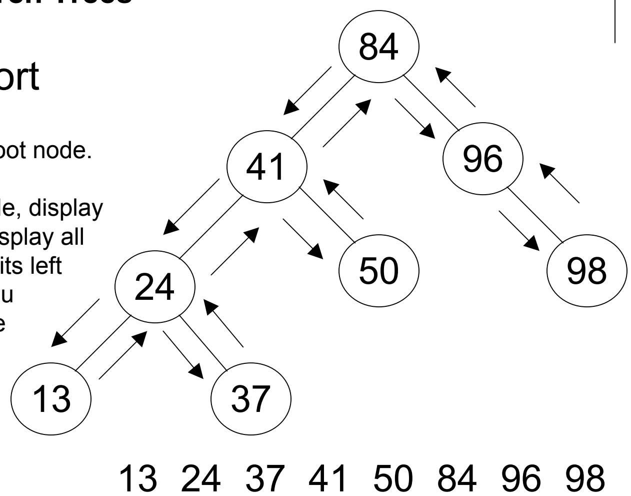


Example
Binary Search Trees

- Tree Sort

Start at the root node.

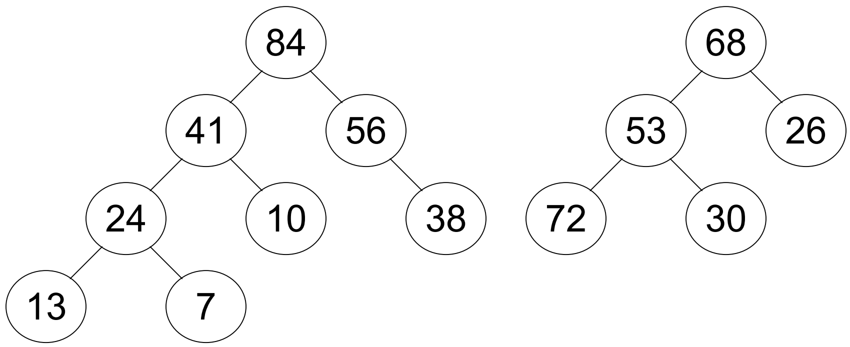
For each node, display it after you display all the nodes to its left but before you display all the nodes to its right.



Example
Heaps

- A heap is a binary tree such that
  - The largest data value is in the root
  - For every node in the heap, its children contain smaller data.
  - The heap is an almost-complete binary tree.
    - An <u>almost-complete binary tree</u> is a binary tree such that every level of the tree has the maximum number of nodes possible except possibly the last level, where its nodes are attached as far left as possible.

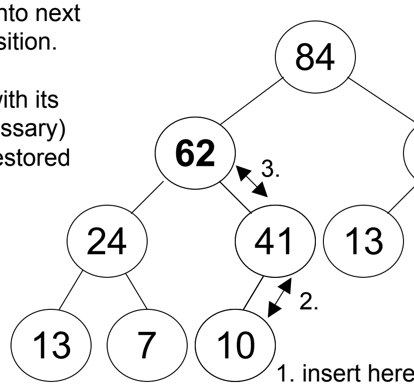
These are not heaps! Why?



Adding data to a heap

Insert new data into next available tree position.

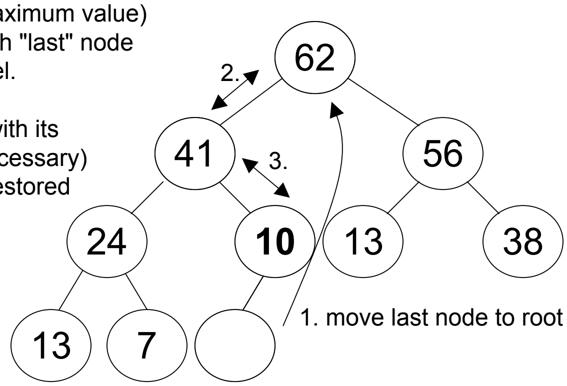
Exchange data with its parent(s) (if necessary) until the tree is restored to a heap.



Removing data from a heap

Remove root (maximum value) and replace it with "last" node of the lowest level.

Exchange data with its larger child (if necessary) until the tree is restored to a heap.



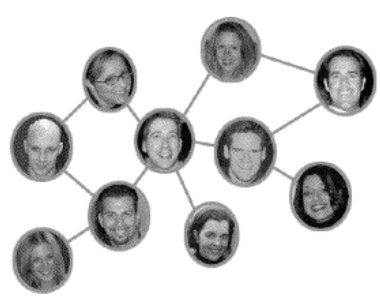
BSTs vs. Heaps

- Which tree is designed for easier searching?
- Which tree is designed for retrieving the maximum value?
- A heap is guaranteed to be a complete or almost-complete tree. What about a BST?

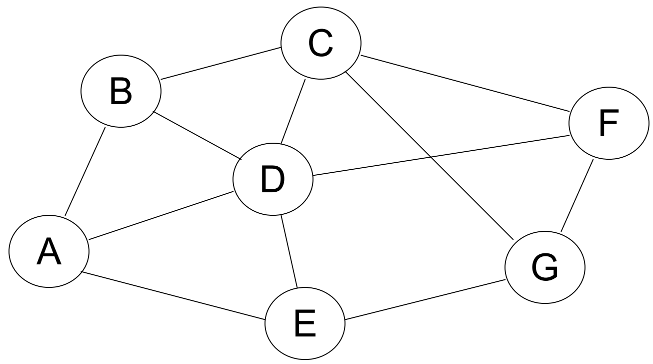
Graphs

- A **graph** is a data structure that consists of a set of nodes and a set of edges connecting pairs of the nodes.
  - A graph doesn’t have a root, per se.
  - A node can be connected to any number of other nodes using edges.
  - An edge may be bidirectional or directed (one-way).
  - An edge may have a weight on it that indicates a cost for traveling over that edge in the graph.
- Applications: computer networks, transportation systems, social relationships

Graphs



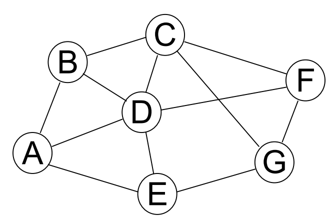
A graph



Storing a graph

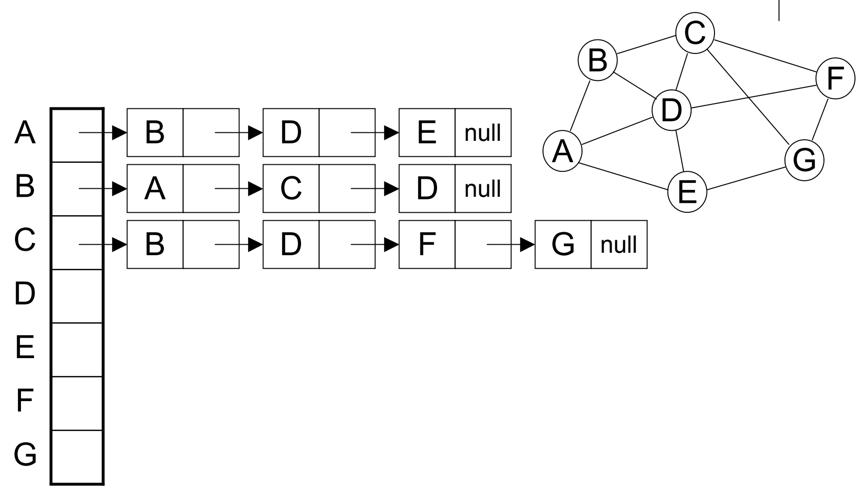
Adjacency Matrix

	A	B	C	D	E	F	G
A	0	1	0	1	1	0	0
B	1	0	1	1	0	0	0
C	0	1	0	1	0	1	1
D							
E							
F							
G							

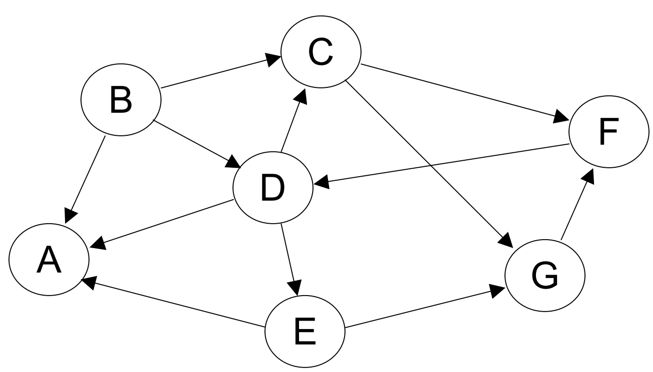


Storing a graph

Adjacency Lists

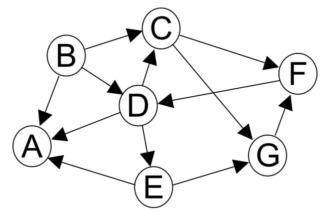


A directed graph

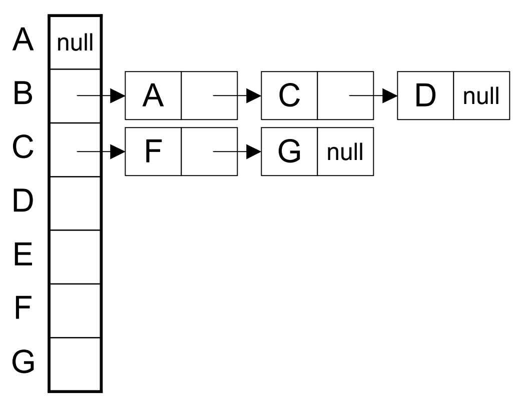


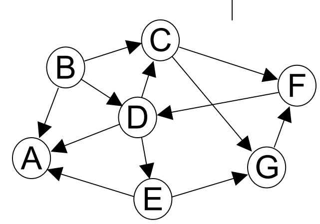
Storing a graph
Adjacency Matrix

	A	B	C	D	E	F	G
A	0	0	0	0	0	0	0
B	1	0	1	1	0	0	0
C	0	0	0	0	0	1	1
D							
E							
F							
G							

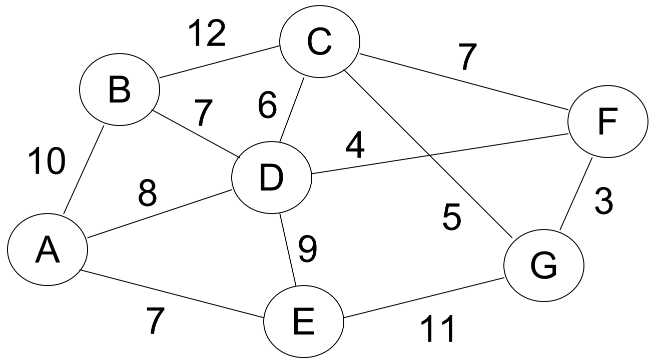


Storing a graph
Adjacency Lists





A weighted graph

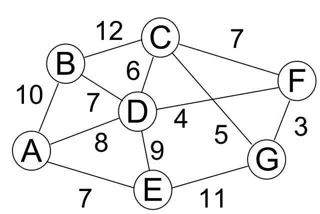


Can you think of examples in real life that can be modeled as weighted graphs?

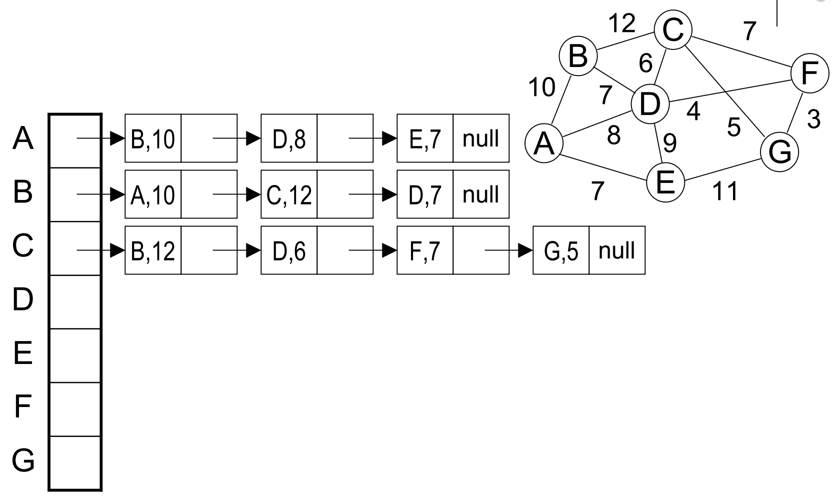
Storing a graph

Adjacency Matrix

	A	B	C	D	E	F	G
A	0	10	0	8	7	0	0
B	10	0	12	7	0	0	0
C	0	12	0	6	0	7	5
D							
E							
F							
G							



Storing a graph
Adjacency Lists

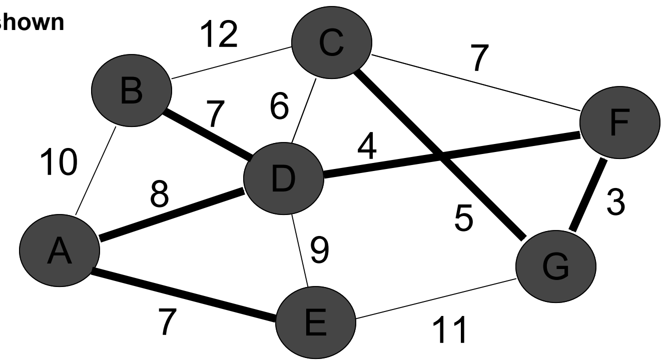


Graph Example
Minimal Spanning Tree (MST)

- Find a tree T in the graph G consisting of the minimum set of edges that connects all the nodes without cycles that has minimum total cost.
- Algorithm:
  - Choose the lowest cost edge of graph G for tree T
  - While all nodes are not part of T do the following:
    - Choose the lowest cost edge that connects with a node currently in the tree T
    - If this edge does not form a cycle in T, add it to T
    - Otherwise, remove this edge from consideration
  - T is a Minimal Spanning Tree of graph G

MST Trace

MST shown in red

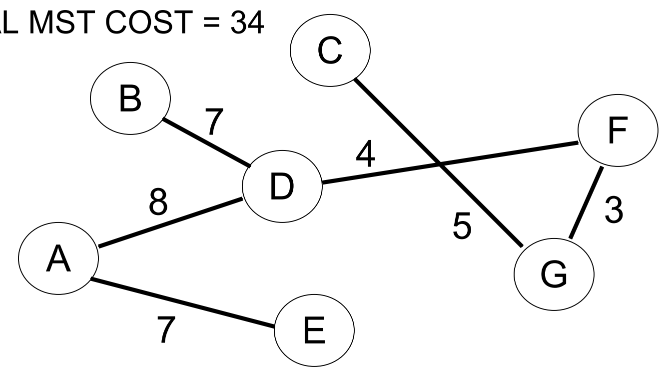


(F,G) 3	(D,F) 4	(C,G) 5	(C,D) -	(B,D) 7	(C,F) -	(A,D) 8	(A,E) 7

Order of edges examined

MST Results

TOTAL MST COST = 34



How did we detect if we were going to put a cycle in the tree?

Databases

- A **database** is a large collection of data.
  - A **schema** describes the data values that are stored in the database, and the relationships that exist between these data values.
  - The computer program used to manage and query a database is known as a **database management system** (DBMS).
    Examples: Oracle, Microsoft SQL Server

Database Models

- Hierarchical
  - links records together in a tree data structure such that each record type has only one owner
  - Example: An order has only one customer.
- Network
  - links records together in a graph data structure such that each record type can have more than owner
  - Example: A student can have more than one professor.

Database Models (cont'd)

- Relational
  - links records together in a matrix based on relationships
  - Example Relations(Fields):
    - Customer(Customer_ID, Name, Address, City, State, Zip, Phone)
    - Order(Customer_ID, Product_Number, Qty_Ordered)
    - Invoice(Invoice_No, Product_Number, Qty_Shipped)
  - Languages like SQL (Structured Query Language) can be used to write this query.

Find all customer names from Pittsburgh who ordered product #15105 but did not receive a full shipment yet.

```
SELECT Name FROM Customer
WHERE Customer.City = "Pittsburgh"
AND Customer.Customer_ID=Order.Customer_ID
AND Order.Product_Number  = 15105
AND Invoice.Product_Number = 15105
AND Order.Qty_Ordered > Invoice.Qty_Shipped
```


Dealing with Huge Data Sets

- Data Mining
  - A procedure where computational techniques such as statistics and pattern matching are used to extract useful information from very large sets of data.
- Data Warehousing
  - A technique that is designed to provide efficient storage for large amounts of data that change infrequently or in localized ways.