Hash tables

$h_0(x) = (h(x) + f_0) mod\, m,\ \ f_i = i$ # Linear probing

$f_i = ih^+(x),\ \ h^+(x) = R - (x\, mod\, R),\ \ \ prime\, R < m$

$f_i = i^2$ # Quadratic probing

For quadratic probing hash table of size $m > 3$, first $\lfloor \frac{m}{2} \rfloor$ probes would be distinct.

$\lambda = \frac{m}{n}$ # Load factor

Ideal: prime m, $\lambda = 1$ for open hashing, $\lambda = \frac{1}{2}$ for closed hashing.

Use separate chaining (linked list) when don't know # of insertion/deletion
Else, use closed hashing

Growth rates:

$if\ \lim_{n\to\infty} \frac{f(n)}{g(n)} = c$, then:

$$0 \le c < \infty \Rightarrow f(n) = O(g(n))$$
$$0 < c \le \infty \Rightarrow f(n) = \Omega(g(n))$$
$$0 < c < \infty \Rightarrow f(n) = \Theta(g(n))$$

$c = 0 \Rightarrow f(n) = o(g(n))$
$c = \infty \Rightarrow f(n) = \omega(g(n))$
Use L'Hopital's rule in the proof

$$\sum_{1\le i\le n} i = \frac{n(n+1)}{2}$$

$$\sum_{1\le i\le n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{1\le i\le n} i^3 = \left(\frac{n(n+1)}{2}\right)^2$$

$$T(n) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} + \sum_{k=1}^{n}\right)C = C\sum_{i=1}^{n}(i + n) = C\left(\frac{n(n+1)}{2} + n^2\right)$$

Trees

Complete binary tree is left-justified

For el i in array binary tree: parent $\frac{i-1}{2}$, left $2i + 1$, right $2i + 2$

Binary Search Trees

$left\ nodes < root \leq right\ nodes$

Inorder traversal = sorted order

Can deserialize preorder traversal

Can balance by rebuilding from inorder serialization

$c_{i,j} = \min_{i \leq k \leq j} \{c_{i,k-1} + c_{k+1,j} + \sum_{l=i}^{j} p_l\}$ # min avg search cost

$c_{i,i} = p_i$

$c_{i+1,i} = 0$

Approach to compute $c_{1,n}$:

1. Compute $c_{i,i}\ for\ all\ i$
2. Compute $c_{i,j}$ in increasing difference of $(j - i)$

$t_{i,j} = k \Leftrightarrow x_k$ is the root of optimal BST

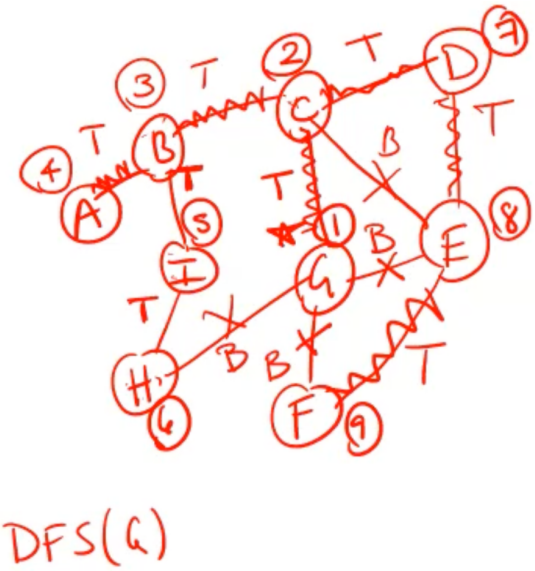
$\{x_i, x_{i+1}, \ldots, x_k, x_{k+1}, \ldots, x_j\}$

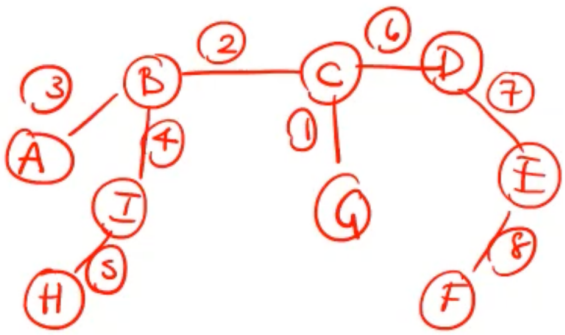
$for\ i = 1\ to\ n\ do$:

$c_{i,i} = p_i$

$t_{i,i} = i$

$Cost = \sum_{i=1}^{n} p_i(d_{epth\ i} + 1)$

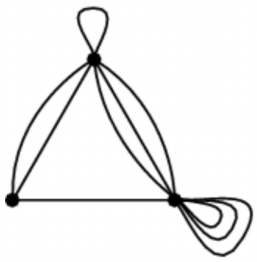




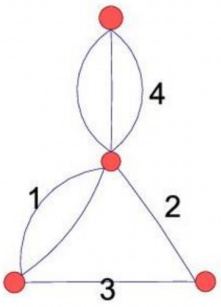
Kruskals MST: add all edges from cheapest skipping loops
Prims MST: store and update costs of neighbours

Additional Graph Definitions

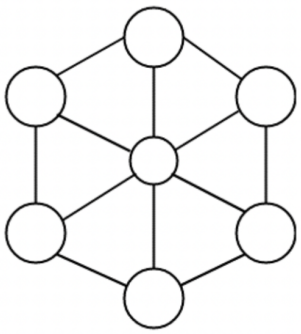
- **Pseudograph:** A graph with possible loops and parallel edges.
- **Multigraph:** A graph with no loops but possible parallel edges.
- **Simple graph:** A graph with no loops or parallel edges.



pseudograph



multigraph



simple graph

More on Connectivity in Undirected Graphs

A vertex v is an **isolated vertex** if it is not connected to any other vertex in V.

G is a **connected graph** iff $\forall x, y \in V$, x and y are connected. Otherwise, it is a **disconnected graph**.

A path is **simple** if all the vertices on the path, except possibly the initial and terminal vertices, are distinct.

Given an undirected graph G = (V, E).

G is a **tree** iff G is a connected graph without a cycle.

G is a **complete graph** iff $\forall x, y \in V$, $(x,y) \in E$.

A tree has n-1 edges, where n is the number of vertices.

A complete graph has n(n-1)/2 edges, where n is the number of vertices.

Connectivity in Directed Graphs

Given a directed graph G = (V, E).

For any vertices $x, y \in V$, a **weakly connected path** from x to y of length k is a sequence of k+1 vertices $(v_0, v_1, \ldots, v_k)$ such that $v_0 = x$, $v_k = y$, and $(v_i, v_{i+1}) \in E$, or $(v_{i+1}, v_i) \in E$, $\forall i, 0 \leq i \leq k\text{-}1$.

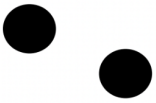
Vertices x and y are **weakly connected** iff there exists a weakly connected path of length k, $k \geq 0$, from x to y.

If $\forall i, 0 \leq i \leq k\text{-}1$, $(v_i, v_{i+1}) \in E$, the path is a **strongly connected path** from x to y of length k.

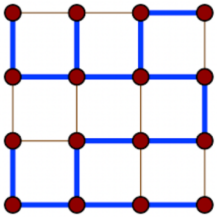
Vertices x and y are **strongly connected** iff there exists a strongly connected path of length k, $k \geq 0$, from x to y.

Some Special Undirected Graphs

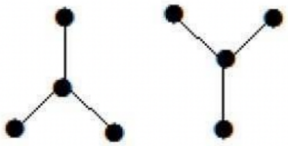
1. A graph with no edge is an **isolated graph**.
2. A connected graph with n vertices and n-1 edges is a **tree**. Equivalently, a tree is a simple connected graph with no cycle.
3. A collection of trees is a **forest**.
4. Given a connected graph G = (V, E). A **spanning tree** of G is any spanning subgraph T = (V, E_T) such that T forms a tree.



isolated graph

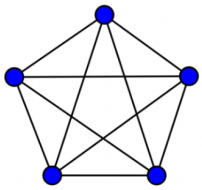


spanning tree



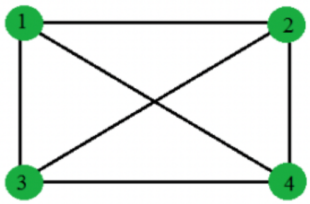
two trees in a forest

Examples of Special Types of Graphs



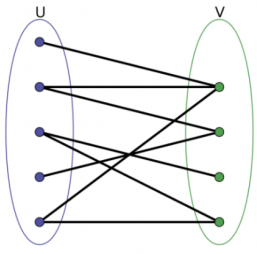
complete graph

K_5

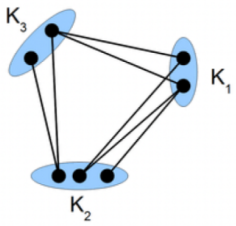


3 Regular

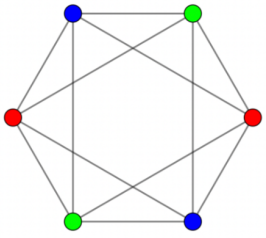
regular graph



bipartite graph



k-partite graph



complete k-partite graph: $K_{2,2,2}$

B-tree Definition

Definition: A **B-tree of order M** is an M-ary tree with the following properties:

1. **All leaf nodes** are on the **same level**.
2. All **non-leaf nodes** (except the root) have **at most M** and **at least** $\lfloor M/2 \rfloor$ **children**.
3. The **number of keys** is **one less than the number of children** for **non-leaf nodes** and **at most M-1** and **at least** $\lfloor M/2 \rfloor$ for leaf nodes.
4. The **root** may have **as few as 2 children** unless the tree is the root alone. It can have **at most M children**.
5. Typically, the number of keys in a node ranges from **d to 2d**, where **2d+1 = M (for M = odd)**.

Note: The **keys** are **sorted**, just like in a BST.

Note: This definition is not consistent across references. Everyone must use this definition.