



CSE408

Fundamentals of Data Structure

Lecture #2

- ☼ list
 - array
 - linked list
 - string
- ☼ stack
- ☼ queue
- ☼ priority queue/heap
- ☼ graph
- ☼ tree and binary tree
- ☼ set and dictionary

⚙ Arrays

- A sequence of n items of the same data type that are stored contiguously in computer memory and made accessible by specifying a value of the array's index.

⚙ Linked List

- A sequence of zero or more nodes each containing two kinds of information: some data and one or more links called pointers to other nodes of the linked list.
- Singly linked list (next pointer)
- Doubly linked list (next + previous pointers)

■ Arrays

- fixed length (need preliminary reservation of memory)
- contiguous memory locations
- direct access
- Insert/delete

Linked Lists

- dynamic length
- arbitrary memory locations
- access by following links
- Insert/delete

⚙ Stacks

- A stack of plates
 - insertion/deletion can be done only at the top.
 - LIFO
- Two operations (push and pop)

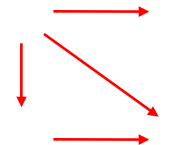
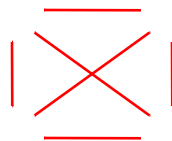
⚙ Queues

- A queue of customers waiting for services
 - Insertion/enqueue from the rear and deletion/dequeue from the front.
 - FIFO
- Two operations (enqueue and dequeue)

- Priority queues (implemented using heaps)
 - A data structure for maintaining a set of elements, each associated with a key/priority, with the following operations
 - Finding the element with the highest priority
 - Deleting the element with the highest priority
 - Inserting a new element
 - Scheduling jobs on a shared computer



- Formal definition
 - A graph $G = \langle V, E \rangle$ is defined by a pair of two sets: a finite set V of items called **vertices** and a set E of vertex pairs called **edges**.
- Undirected** and **directed** graphs (**digraphs**).
- What's the maximum number of edges in an undirected graph with $|V|$ vertices?
- Complete, dense, and sparse** graphs
 - A graph with every pair of its vertices connected by an edge is called complete, $K_{|V|}$



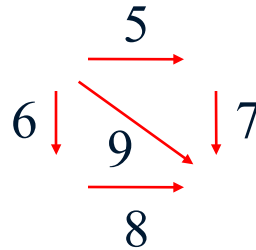
- ⊗ Adjacency matrix
 - $n \times n$ boolean matrix if $|V|$ is n .
 - The element on the i th row and j th column is 1 if there's an edge from i th vertex to the j th vertex; otherwise 0.
 - The adjacency matrix of an undirected graph is symmetric.
- ⊗ Adjacency linked lists
 - A collection of linked lists, one for each vertex, that contain all the vertices adjacent to the list's vertex.
- ⊗ Which data structure would you use if the graph is a 100-node star shape?

0	1	1	1
0	0	0	1
0	0	0	1
0	0	0	0



⊗ Weighted graphs

- Graphs or digraphs with numbers assigned to the edges.



⊗ Paths

- A path from vertex u to v of a graph G is defined as a sequence of adjacent (connected by an edge) vertices that starts with u and ends with v .
- **Simple paths:** All edges of a path are distinct.
- Path lengths: the number of edges, or the number of vertices $- 1$.

⊗ Connected graphs

- A graph is said to be connected if for every pair of its vertices u and v there is a path from u to v .

⊗ Connected component

- The maximum connected subgraph of a given graph.

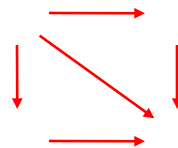


⊗ Cycle

- A simple path of a positive length that starts and ends at the same vertex.

⊗ Acyclic graph

- A graph without cycles
- **DAG** (Directed Acyclic Graph)

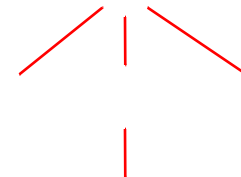


⊗ Trees

- A tree (or **free tree**) is a connected acyclic graph.
- Forest: a graph that has no cycles but is not necessarily connected.

⊗ Properties of trees

- For every two vertices in a tree there always exists exactly one simple path from one of these vertices to the other. **Why?**
 - **Rooted trees:** The above property makes it possible to select an arbitrary vertex in a free tree and consider it as the root of the so called rooted tree.
 - Levels in a rooted tree.



• Ancestors

- For any vertex v in a tree T , all the vertices on the simple path from the root to that vertex are called ancestors.

• Descendants

- All the vertices for which a vertex v is an ancestor are said to be descendants of v .

• Parent, child and siblings

- If (u, v) is the last edge of the simple path from the root to vertex v , u is said to be the parent of v and v is called a child of u .
- Vertices that have the same parent are called siblings.

• Leaves

- A vertex without children is called a leaf.

• Subtree

- A vertex v with all its descendants is called the subtree of T rooted at v .

Rooted Trees (II)

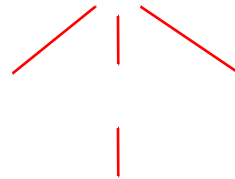


☼ Depth of a vertex

- The length of the simple path from the root to the vertex.

☼ Height of a tree

- The length of the longest simple path from the root to a leaf.



- ⊗ Ordered trees
 - An ordered tree is a rooted tree in which all the children of each vertex are ordered.
- ⊗ Binary trees
 - A binary tree is an ordered tree in which every vertex has no more than two children and each children is designated as either a left child or a right child of its parent.
- ⊗ Binary search trees
 - Each vertex is assigned a number.
 - A number assigned to each parental vertex is larger than all the numbers in its left subtree and smaller than all the numbers in its right subtree.
- ⊗ $\lfloor \log_2 n \rfloor \leq h \leq n - 1$, where h is the height of a binary tree and n the size.





Thank You !!!