Design and Analysis of Algorithms

L08: Fundamental Data Structures

Dr. Ram P Rustagi
Sem IV (2020-Even)
Dept of CSE, KSIT/KSSEM
rprustagi@ksit.edu.in

Resources

- Text book 1: Levitin

Fundamental Data Structures

- Primarily support 4 kinds of operations
  - Insert (or add) an item
  - Search (or find) an item; Find min is specific case
  - Delete an item; Delete min is specific case
  - Modify (or update)
- Cost (efficiency) of the operation depends on underlying data structure in use.
- Unsorted array:
  - Insert: O(`1`)
  - Search O(`n`)
- Sorted array
  - Insert: O(`n`)
  - Search: O(`log n`)

Fundamental Data Structures

- Choosing a data structure
  - Determine the operations you need to perform,
  - How much cost to be paid for operation
- Examples:
  - Email:
    - Insert, delete, search
    - No modify/update
  - Class attendance
    - Insert (fastest), modify and search (rarely),
    - No delete operation
  - Contacts (or address book)
    - All 4 operations
    - Search should be fastest

Fundamental Data Structures

- Lists
  - Arrays, Linked Lists, Strings
- Stacks
- Queues
- Priority queues
- Trees and Binary trees
- Graphs
- Sets
- Dictionaries

Arrays

- Arrays
  - A sequence of `n` items of same types
  - Stored continguously in memory
  - Elements are accessed by element index
  - Single dimensional, multi-dimensional array
- Operations on Arrays
  - Read
  - Add
  - Remove
  - Modify

Lists

- Linked List
  - A sequence of `n` items (nodes)
  - Node has two kind of information
    - Some data corresponding to node
    - One or more links to other nodes
  - Singly linked list
  - Doubly linked list
- Operations on linked lists
  - Add (front of queue)
  - Remove (tail of queue)
  - Search (traverse the queue)
  - Modify (search and update)

Stacks

- Storing items in a way that only top item is accessible
  - Also called LIFO
- Operations:
  - Push (Add)
  - Pop (Delete)
  - Read
  - Search ?? (not inside the stack)
  - IsEmpty
  - IsFull?
- Typically implemented as array (or even list)

Queues

- Storing items in a way that only first (head) and last (tail) item is accessible
  - Also called FIFO
- Operations:
  - Enqueue (Add)
    - At the rear
  - Dequeue (Delete)
    - At the front

Priority Queues

- A data structure that maintains items (elements) such that each is associated with
  – A key (or priority) value
- Operations
  – Finding item with highest priority (find min)
  – Deleting the item with highest priority (delete min)
  – Inserting a new element (with its own priority)
- Examples:
  – Scheduling a job on computer

Graphs

- A graph $G = <V, E>$ 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.
- Graphs are of two types
  - Undirected Graphs
    - Given a graph of n nodes, max edges ?
  - Directed graphs
- Other graph categorization
  - Complete Graph
  - Dense Graph
  - Sparse graphs

Graph Representation

- Adjacency matrix
  - n x 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;
    - contain all the vertices adjacent to the list's vertex.
- Weighted graphs: edges with weights
- Q: which data structure would you use if the graph is a 100-node star shape?

Graph Properties

- Path from node u to v
  - A sequence of adjacent (connected by an edge) vertices that starts with u and ends with v.
  - Simple path: all edges of a path are distinct.
  - Q: what happens when edges are not distinct?
- Connected graphs
  - For every pair of its vertices u and v
    - There exists a path from u to v.
- Subgraph
  - A subset V' of V, with all of its edge corresponding to V'
    - if $u \in V'$, $v \in V'$, and $(u,v) \in E$, then $(u, v) \in E'$
- Connected component
  - The maximum connected subgraph of a given graph.
- Strongly connected components (for directed graphs)

Graphs: Acyclicity

- Cycles in a graph G=(V,E)
  - A simple path of positive length that starts from a vertex and ends at same vertex
- Cyclic graphs
  - A graph having cycles
- Acyclic graphs
  - A graphh without cycles
  - Direced acyclic graphs

Trees & Forest

- Tree is a connected acyclic graph
- Forest: A graph that has no cycles but necessarily not connected
  - There may exist 2 nodes u and v for which no path exist between them
- Every two vertices of tree
  - There exists exactly one path between these nodes
- Rooted tree:
  - Identify a vertex of tree and deginate is as root
  - Levels in a rooted tree
    - Root is level 0,
    - Directly connected nodes from root are at level 1.

Rooted Trees

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

Rooted Trees

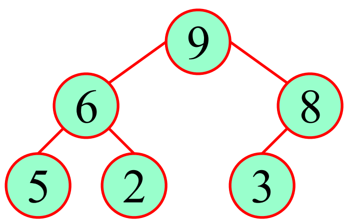
- Subtree
  - A vertex v with all its descendants is called the subtree of T rooted at v.
- Depth of a vertex
  - The length of the simple path from the root to the vertex.
- Height of a tree
  - The length of the longest simple path from the root to a leaf.

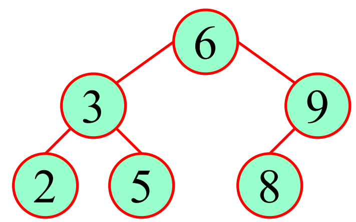
Ordered Trees

- Ordered trees
  - An ordered tree is a rooted tree, where
    - All the children of each vertex are ordered.
- Binary tree: an ordered tree in which every vertex
    - has max 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.

Ordered Trees

- Q: Which of the following tree is ordered tree and which one is binary search tree?





Summary

- Lists
- Stacks
- Queues
- Priority Queues
- Graphs
- Trees
- Ordered trees
- Sets
- Dictionaries