

Notes on Trees

General trees not necessarily rooted

Rooted trees are directed away from root

m-ary = m or fewer children per node

level = distance from root. Child has level 1

height = max(level)

Regular m-ary tree = exactly m or 0 children

Complete = regular and leaves at same level

Almost compl. = maybe miss nodes on bottom

Notes on Graphs

= a pictorial way to represent information

A graph must have at least one vertex, but can have zero edges

It can have vertices that are not connected to anything (disconnected)

Adjacent vertices -> B is adjacent to A if there is an arrow connecting them and pointing at B

Path of length n -> sequence of n edges between two vertices

Simple path = no repeats

Cycle aka closed path = begins and ends at same vertex

Acyclic graph has no cycles

Edges can have weights

Degree = number of incident edges. Also has in-degree and out-degree

If in-degree = 0 -> is source

If out-degree = 0 -> is sink

If every vertex has same degree -> graph is regular

|  |  |  |
| --- | --- | --- |
| Structure | Required | Optional |
| Stack | push, pop, isEmpty | copy, display, peek, clear |
| Queue | insert, delete, isEmpty | copy, peek |
| List, SortedList | insert, delete | copy, display, search |
| BinaryTree | makeTree, setLetf, setRight, traverse, search, delNode | display |