Prerequisites for CSC 349

Powers of 2 : $2^0=1, 2^1=2, 2^2=4, 2^3=8, 2^4=16, 2^5=32, 2^6=64, 2^7=128, 2^8=256, 2^9=512, 2^{10}=1024$

Properties of Exponents: $x^a * x^b = x^{a+b}$; $x^a / x^b = x^{a-b}$; $(x^a)^b = x^{a*b}$;

Properties of Logarithm: In this course all log will be base 2 unless otherwise noted
log (x*y) = log(x) + log(y); log (x/y) = log(x) - log(y); log (x^y) = y* log(x);
and the very important identity for asymptotic analysis that says that log functions for different bases differ bu a constant factor: $\log_a(x) = \mathbf{\log_a(b)} * \log_b(x)$

Important Summation Formulas

$\sum_{i=1}^{n} i = \frac{n*(n+1)}{2}$ This generalizes to $\sum_{i=1}^{n} Ci$ =C $\frac{n*(n+1)}{2}$

$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ a≠ 1 when a =2 then $\sum_{i=0}^{n} 2^i = \frac{2^{n+1}-1}{2-1} = 2^{n+1} - 1$

Definitions of Floor and Ceiling Functions

The **floor** function: $\lfloor x \rfloor$ = greatest integer less than or equal to x . E.g. $\lfloor 4 \rfloor = 4$, $\lfloor 2.34 \rfloor = 2$, $\lfloor -2.2 \rfloor = -3$

The **ceiling** function: $\lceil x \rceil$ = least integer greater than or equal to x. E.g: $\lceil 4 \rceil = 4$, $\lceil 2.34 \rceil = 3$, $\lceil -2.2 \rceil = -2$

Simple recurrence relations and their solution

A **sequence** is an (may be infinite) ordered list of elements. (The elements of a sequence are typically labeled by some subset of the integers.) Simple examples are: The sequence of positive integers: 1, 2, 3, 4, . . . , n, .. The sequence of powers of 2: 1, 2, 4, 8, 16, . . . , 2^n,

The elements of a sequence are represented with an indexed letter, say $a_1, a_2, a_3, \ldots, a_n, \ldots$

The sequence itself can be defined by giving a rule, e.g.: $a_n = 2n + 1$ is the sequence: 3, 5, 7, 9, . . .

The sequence is symbolically represented $\{a_n\}$ or $\{a_n\}_{n=1}^{\infty}$.

Sequences can be defined by a ***recurrence relation***.

1, 2, 4, 8, 16, . . . , 2^n, . . . can be defined by $a_n = 2a_{n-1}$ for n>1 and $a_0 = 1$; each term is computed using earlier terms. E.g. n! = n*(n-1)! and a base case 0! = 1

Solving recurrence relations associated with elementary algorithms

http://www.radford.edu/~nokie/classes/360/recurrence.eqns.revised.html

Analysis of Algorithms

The following concepts have proven to be useful in both the practical and theoretical understanding of the performance of algorithms. This course will use these concepts throughout the course.

- *Big Oh Notation*. A function f(n) is said to be of order *at most g(n)* or order *Big-O of g(n)* , written $f(n) \in O(g(n))$, if there is a constant C_1 such that $|f(n)| \le C_1|g(n)|$ for all but finitely many positive integers n.
- *Big Omega Notation*. A function f(n) is said to be of order *at least g(n)* or order *Big-Ω of g(n)*, written $f(n) \in \Omega(g(n))$, if there is a constant C_2 such that $|f(n)| \ge C_2|g(n)|$ for all but finitely many positive integers n.
- *Big Theta Notation*. A function f(n) is said to be of order g(n) or order *Big-Θ of g(n)*, written $f(n) \in \Theta(g(n))$, if $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$.

Again all logarithmic functions are of the same order: $\log_a n = \_(\log_b n)$ for any a, b > 1, because $\log_a n = \log_b n / \log_b a$, so they always differ in a multiplicative constant. As a consequence, if the execution time of an algorithm is of order a logarithmic function, we can just say that its time is "logarithmic" we do not need to specify the base of the logarithm.

The following are several common growth functions and their names

Order	Name	Order	Name
$\Theta(1)$	Constant	$\Theta(n^2)$	Quadratic
$\Theta(\log n)$	Logarithmic	$\Theta(n^3)$	Cubic
$\Theta(n \log n)$	n log n	$\Theta(n^k)$	Polynomial
$\Theta(n)$	Linear	$\Theta(a^n)$	Exponential

Basic definitions and facts about Graphs

A *graph G* consists of two sets:

1. A *vertex set* V , whose elements are called vertices, nodes or points.
2. An *edge set E* or set of edges connecting pairs of vertices. If the edges are directed then they are also called directed edges or arcs. Each edge $e \in E$ is associated with a pair of vertices and if the edges are directed the graph is called a *directed graph*.

A graph is represented by the pair (V,E) (we assume V and E finite), this is written G= (V,E).

A graph is ***undirected*** if the pairs are ***unordered pairs*** of vertices. If there is a unique edge e connecting x and y we may write e = {x, y}, so E can be regarded as set of *unordered pairs*. In this context we may also write e = (x, y), understanding that here (x, y) is not an ordered pair, but the name of an edge.

A graph is ***directed*** if the pairs are ***ordered pairs*** **of vertices**. If a graph is *directed* and there is a unique edge e pointing from x to y, then we may write e = (x, y), so E may be regarded as a set of ordered pairs. If e = (x, y), the vertex x is called origin, source or initial point of the edge e, and y is called the terminus, terminating vertex or terminal point.

Two vertices connected by an edge are called ***adjacent***. They are also the endpoints of the edge, and the edge is said to be ***incident*** to each of its endpoints. If the graph is directed, an edge pointing from vertex x to vertex y is said to be ***incident*** from x and incident to y. An edge connecting a vertex to itself is called a loop. Two edges connecting the same pair of points (and pointing in the same direction if the graph is directed) are called parallel or multiple. A graph with neither loops nor multiple edges is called a ***simple graph***. **Most of the graphs in this course will be simple graphs.**

The ***degree of a vertex v***, represented **deg(v),** is the number of edges that contain it (loops are counted twice). A vertex of degree zero (not connected to any other vertex) is called isolated. **In-degree of a vertex** v is the number of edges with v as the terminal vertex and the **out degree of a vertex** v is the number with v as the initial vertex.
Complete Graph is a graph with an edge between each pair of vertices

Paths and Circuits

A ***path*** from v_0 to v_n of length n is a sequence of n+1 vertices (v_k) and n edges (e_k) of the form v_0, e_1, v_1, e_2, v_2, . . . , e_n, v_n, where each edge e_k connects v_{k-1} with v_k (and points from v_{k-1} to v_k if the edge is directed). The path may be specified by giving only the sequence of edges e_1, e_2, …, e_n. If there are no multiple edges we can specify the path by giving only the vertices: v_0, v_1, . . . , v_n.
The path is a **circuit (or cycle)** if it begins and ends at the same vertex, i.e., $v_0 = v_n$, and has length greater than zero. A path or circuit is ***simple*** *if it does not contain the same edge twice*.

Prerequisites for CSC 349

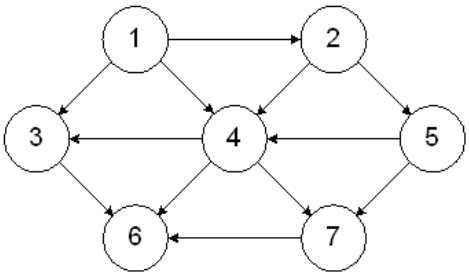
Connected Graphs.

An **undirected graph G is called connected** if there is a path between any two distinct vertices of G. Otherwise the graph is called disconnected. A **directed graph is (weakly) connected** if its associated undirected graph (obtained by ignoring the directions of the edges) is connected. A **directed graph is strongly connected** if for every pair of distinct points u, v, there is a path from u to v and there is a path from v to u.

A **connected component of G** is any connected subgraph $G_0 = (V_0, E_0)$ of $G = (V,E)$ such that there is not edge (in G) from a vertex in V_0 to a vertex in $V - V_0$. Given a vertex in G, the ***component of G containing v*** is the subgraph G_0 of G consisting of all edges and vertices of g contained in some path beginning at v.

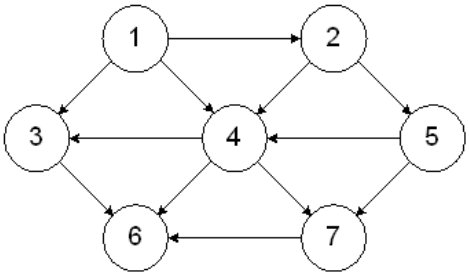
Graph Representations

Adjacency matrix. The adjacency matrix of a graph is a matrix with rows and columns labeled by the vertices and such that its entry in row i, column j, is the number of edges incident on i and j.



	[1]	[2]	[3]	[4]	[5]	[6]	[7]
[1]	0	1	1	1	0	0	0
[2]	0	0	0	1	1	0	0
[3]	0	0	0	0	0	1	0
[4]	0	0	1	0	0	1	1
[5]	0	0	0	1	0	0	1
[6]	0	0	0	0	0	0	0
[7]	0	0	0	0	0	1	0

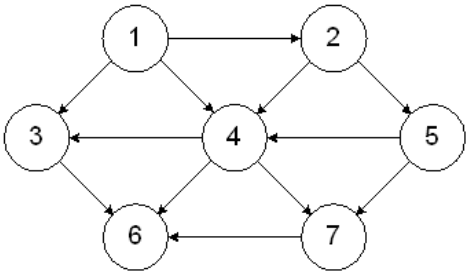
A directed graph and adjacency matrix

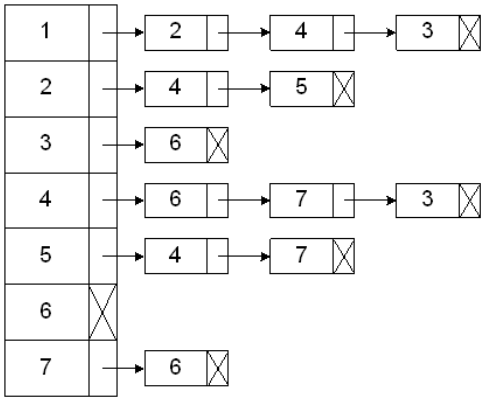


	[1]	[2]	[3]	[4]	[5]	[6]	[7]
[1]	0	1	1	1	0	0	0
[2]	1	0	0	1	1	0	0
[3]	1	0	0	1	0	1	0
[4]	1	1	1	0	1	1	1
[5]	0	1	0	1	0	0	1
[6]	0	0	1	1	0	0	1
[7]	0	0	0	1	1	1	0

An undirected (ignore the arrow heads) graph and adjacency matrix

The space requirement for an adjacency matrix is V^2, where V is the number of vertices. To avoid this problem, a graph can also be represented as an adjacency list. For this type of representation, an array is used to hold the data for each vertex in the graph. For each vertex, there is a list of all vertices that are adjacent to the vertex.





A directed graph and an adjacency list **Source http://faculty.cs.niu.edu**

The space requirement for an adjacency list is E+V, where E is the number of edges and V is the number of vertices.

Def: An undirected graph is a tree if there is a unique simple path between any pair of vertices.

A tree is a simple undirected graph that satisfies any of the following three conditions:

- It is connected and |E| = |V| - 1
- It is connected and has no simple circuits (acyclic)
- It is acyclic and |E| = |V| - 1