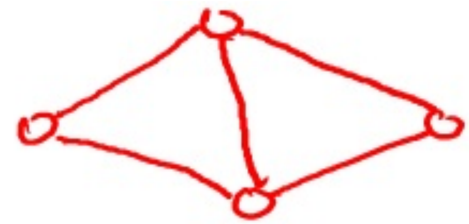
Design and Analysis of Algorithms I

Graph Algorithms

Representing Graphs

Graphs



Two ingredients

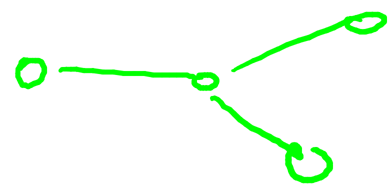
- Vertices aka nodes (V)
- Edges (E) = pairs of vertices
    - can be undirected [unordered pair] or directed [ordered pair] (aka arcs)

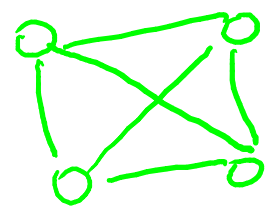
Examples: road networks, the Web, social networks, precedence constraints, etc.

Ter
ver

Consider an undirected graph that has n vertices, no parallel edges, and is connected (i.e., "in one piece"). What is the minimum and maximum number of edges that the graph could have, respectively ?

- $n-1$ and $n(n-1)/2$
- $n-1$ and n^2
- n and 2^n
- n and n^n





Sparse vs. Dense Graphs

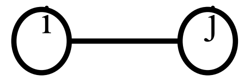
Let <u>n</u> = # of vertices,   <u>m</u> = # of edges.

In most (but not all) applications, m is $\Omega(n)$ and $O(n^2)$

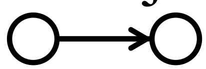
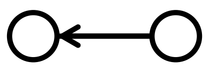
- in a "sparse" graph, m is or is close to $O(n)$
- in a "dense" graph, m is closer to $\theta(n^2)$

Tim Roughgarden

The Adjacency Matrix

Represent G by a n x n 0-1 matrix A where
$A_{ij} = 1 \Leftrightarrow$ G has an i-j edge 

<u>Variants</u>

- A_{ij} = # of i-j edges (if parallel edges)
- A_{ij} = weight of i-j edge (if any)
- A_{ij} = [+1 if 
  -1 if 

Ter
ver

How much space does an adjacency matrix require, as a function of the number n of vertices and the number m of edges?

- $\theta(n)$
- $\theta(m)$
- $\theta(m+n)$
- $\theta(n^2)$

Adjacency Lists

<u>Ingredients</u>

- array (or list) of vertices
- array (or list) of edges
- each edge points to its endpoints
- each vertex points to edges incident on it

Ter
ver

How much space does an adjacency list representation require, as a function of the number n of vertices and the number m of edges?

- $\theta(n)$
- $\theta(m)$
- $\theta(m + n)$
- $\theta(n^2)$

Adjacency Lists

Ingredients	Space
• array (or list) of vertices	$\theta(n)$
• array (or list) of edges	$\theta(m)$
• each edge points to its endpoints	$\theta(m)$
• each vertex points to edges incident on it	$\theta(m)$
	$\theta(m+n)$
	$[or \;\; \theta(max\{m,n\})]$

one-to-one correspondence !

Question: which is better?

Answer: depends on graph density and operations needed.

This course: focus on adjacency lists.