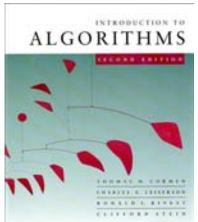
CS 5633 -- Spring 2006



Graphs

Carola Wenk

Slides courtesy of Charles Leiserson with changes and additions by Carola Wenk

Graphs (review)

Definition. A ***directed graph (digraph)*** $G = (V, E)$ is an ordered pair consisting of

- a set V of ***vertices*** (singular: ***vertex***),
- a set $E \subseteq V \times V$ of ***edges***.

In an ***undirected graph*** $G = (V, E)$, the edge set E consists of *unordered* pairs of vertices.

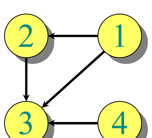
In either case, we have $|E| = O(|V|^2)$. Moreover, if G is connected, then $|E| \geq |V| - 1$.

(Review CLRS, Appendix B.4 and B.5.)

Adjacency-matrix representation

The ***adjacency matrix*** of a graph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$, is the matrix $A[1 \,.\,.\, n, 1 \,.\,.\, n]$ given by

$$A[i, j] = \begin{cases} 1 & \text{if } (i, j) \in E, \\ 0 & \text{if } (i, j) \notin E. \end{cases}$$

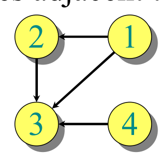


A	1	2	3	4
1	0	1	1	0
2	0	0	1	0
3	0	0	0	0
4	0	0	1	0

$\Theta(|V|^2)$ storage $\Rightarrow$ ***dense*** representation.

Adjacency-list representation

An ***adjacency list*** of a vertex $v \in V$ is the list $Adj[v]$ of vertices adjacent to v.



$Adj[1] = \{2, 3\}$
$Adj[2] = \{3\}$
$Adj[3] = \{\}$
$Adj[4] = \{3\}$

For undirected graphs, $|Adj[v]| = degree(v)$.

For digraphs, $|Adj[v]| = out\text{-}degree(v)$.

Graph Traversal

Let $G=(V,E)$ be a (directed or undirected) graph, given in adjacency list representation.

$|V| = n$, $|E| = m$

A graph traversal visits every vertex:

- Breadth-first search (BFS)
- Depth-first search (DFS)

Breadth-First Search (BFS)

BFS($G=(V,E)$)
    Unmark all vertices
    Choose some starting vertex s
    Mark s
    queue $Q = s$
    tree $T = s$
    while Q is non-empty **do**
        $v = Q$.dequeue
        visit v
        for each unmarked w adjacent to v **do**
            Mark w
            Q.enqueue(w)
            Add edge (v,w) to T

BFS runtime

- Each vertex is unmarked in the beginning $\Rightarrow$ $O(n)$ time
- Each vertex is marked at most once, enqueued at most once, and therefore dequeued at most once
- The time to process a vertex is proportional to the size of its adjacency list (its degree), since the graph is given in adjacency list representation
  $\Rightarrow$ $O(m)$ time
- Total runtime is $O(m+n)$

Depth-First Search (DFS)

DFS(G, v)
    visit v
    for each w adjacent to v **do**
        if w is unvisited
            DFS(G,w)
            Add edge (v,w) to tree T

DFS runtime

- Each vertex is visited at most once ⇒ O(n) time
- The body of the **for** loop (except the recursive call) takes constant time per graph edge
- All **for** loops take O(m) time
- Total runtime is O(m+n)