Algorithm Design and Analysis

LECTURE 4

- Basic Data Structures
- Graphs

Adam Smith
Computational Model

Unless explicitly stated otherwise

• All numbers and pointers fit into a single word (block) of memory

• Constant-time operations
  – Operations on words: arithmetic op’s, shifts, comparisons, etc
  – Following a pointer
  – Array lookup

We will sometimes drop these assumptions

• E.g.: for numerical problems, we might count bit op’s
Basic Data Structures

- Lists
  - O(1) time: Insert/delete anywhere we have a pointer
- Array
  - O(1) time: append, lookup

**Good for**

- Stack: Last in, First out (LIFO)
  - O(1) time: Push, pop
- Queue: First in, First out (FIFO)
  - O(1) time: enqueue, dequeue
## Dictionary Data Structures

<table>
<thead>
<tr>
<th>Data Struct.</th>
<th>Find</th>
<th>Insert</th>
<th>Delete (after Find)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsorted array</td>
<td>$\Theta(n)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(1)$</td>
</tr>
<tr>
<td>Linked list</td>
<td>$\Theta(n)$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(1)$</td>
</tr>
<tr>
<td>Sorted array</td>
<td>$\Theta(\log(n))$</td>
<td>$\Theta(n)$</td>
<td>$\Theta(n)$</td>
</tr>
<tr>
<td>Binary search tree</td>
<td>$\Theta(height)$</td>
<td>$\Theta(height)$</td>
<td>$\Theta(height)$</td>
</tr>
<tr>
<td>Balanced binary search tree</td>
<td>$\Theta(\log(n))$</td>
<td>$\Theta(\log(n))$</td>
<td>$\Theta(\log(n))$</td>
</tr>
<tr>
<td>Hash table (\text{(expected time over the choice of hash function; worst case over data)})</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Here $n = \#$ of items currently in dictionary.
Table entries are worst-case asymptotic running times.

9/3/2008

*A. Smith; based on slides by E. Demaine, C. Leiserson, S. Raskhodnikova, K. Wayne*
Definition. A directed graph (digraph) \( G = (V, E) \) is an ordered pair consisting of

- a set \( V \) of vertices (synonym: nodes),
- a set \( E \subseteq V \times V \) of edges
- An edge \( e = (u, v) \) goes “from \( u \) to \( v \)” (may or may not allow \( u = v \))

- In an undirected graph \( G = (V, E) \), the edge set \( E \) consists of unordered pairs of vertices
  - Sometimes write \( e = \{u, v\} \)

- How many edges can a graph have?
  - In either case, \( |E| = O(V^2) \).
# Graphs are everywhere

<table>
<thead>
<tr>
<th>Example</th>
<th>Nodes</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transportation network: airline routes</td>
<td>airports</td>
<td>nonstop flights</td>
</tr>
<tr>
<td>Communication networks</td>
<td>computers, hubs, routers</td>
<td>physical wires</td>
</tr>
<tr>
<td>Information network: web</td>
<td>pages</td>
<td>hyperlinks</td>
</tr>
<tr>
<td>Information network: scientific papers</td>
<td>articles</td>
<td>references</td>
</tr>
<tr>
<td>Social networks</td>
<td>people</td>
<td>“u is v’s friend”, “u sends email to v”, “u’s MySpace page links to v”</td>
</tr>
</tbody>
</table>

9/3/2008

*A. Smith; based on slides by E. Demaine, C. Leiserson, S. Raskhodnikova, K. Wayne*
Paths and Connectivity

- **Path** = sequence of consecutive edges in $E$
  - $(u,w_1), (w_1,w_2), (w_2,w_3), \ldots, (w_{k-1}, v)$
  - Write $u \sim v$ or $u \sim v$
  - (Note: in a directed graph, direction matters)

- Undirected graph $G$ is **connected** if for every two vertices $u,v$, there is a path from $u$ to $v$ in $G$

- Directed graph?
  - **Strongly connected** if for every pair, $u \sim v$ and $v \sim u$

A. Smith; based on slides by E. Demaine, C. Leiserson, S. Raskhodnikova, K. Wayne