Lecture 6 – Graphs

Reading: KT Sections 3.1 and 3.2

Partial content of these slides have been obtained from the official lecture slides that accompany the textbook. A complete set of slides can be found at: http://www.cs.princeton.edu/~wayne/kleinberg-tardos/

Let’s start with some basics

Graph representation

Undirected Graphs

- Undirected graph. \( G = (V, E) \)
  - \( V \) = nodes.
  - \( E \) = edges between pairs of nodes.
  - Captures pairwise relationship between objects.
  - Graph size parameters: \( n = |V|, m = |E| \).

\[
V = \{1, 2, 3, 4, 5, 6, 7, 8\}
\]
\[
E = \{1-2, 1-3, 2-3, 2-4, 2-5, 3-5, 3-7, 3-8, 4-5, 5-6\}
\]
\( n = 8 \)
\( m = 11 \)

Some Graph Applications

<table>
<thead>
<tr>
<th>Graph</th>
<th>Nodes</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>transportation</td>
<td>street intersections</td>
<td>highways</td>
</tr>
<tr>
<td>communication</td>
<td>computers</td>
<td>fiber optic cables</td>
</tr>
<tr>
<td>World Wide Web</td>
<td>web pages</td>
<td>hyperlinks</td>
</tr>
<tr>
<td>social</td>
<td>people</td>
<td>relationships</td>
</tr>
<tr>
<td>food web</td>
<td>species</td>
<td>predator-prey</td>
</tr>
<tr>
<td>software systems</td>
<td>functions</td>
<td>function calls</td>
</tr>
<tr>
<td>scheduling</td>
<td>tasks</td>
<td>precedence constraints</td>
</tr>
<tr>
<td>circuits</td>
<td>gates</td>
<td>wires</td>
</tr>
</tbody>
</table>
World Wide Web

- Web graph.
  - Node: web page.
  - Edge: hyperlink from one page to another.

Ecological Food Web

- Food web graph.
  - Node: species.
  - Edge: from prey to predator.

9-11 Terrorist Network

- Social network graph.
  - Node: people.
  - Edge: relationship between two people.


Paths and Connectivity

- Def. A path in an undirected graph $G = (V, E)$ is a sequence $P$ of nodes $v_1, v_2, \ldots, v_k$ with the property that each consecutive pair $v_i, v_{i+1}$ is joined by an edge in $E$.

- Def. A path is simple if all nodes are distinct.

- Def. An undirected graph is connected if for every pair of nodes $u$ and $v$, there is a path between $u$ and $v$. 

Cycles

- Def. A cycle is a path \( v_1, v_2, \ldots, v_{k-1}, v_k \) in which \( v_1 = v_k \), \( k > 2 \), and the first \( k-1 \) nodes are all distinct.

Trees

- Def. An undirected graph is a tree if it is connected and does not contain a cycle.

- Theorem. Let \( G \) be an undirected graph on \( n \) nodes. Any two of the following statements imply the third.
  - \( G \) is connected.
  - \( G \) does not contain a cycle.
  - \( G \) has \( n-1 \) edges.

Rooted Trees

- Rooted tree. Given a tree \( T \), choose a root node \( r \) and orient each edge away from \( r \).

- Importance. Models hierarchical structure.

Exercise time!
Graph Representation: Adjacency Matrix

- Adjacency matrix. n-by-n matrix with $A_{uv} = 1$ if (u, v) is an edge.
- Two representations of each edge.
- Space proportional to _______.
- Checking if (u, v) is an edge takes $\Theta(?)$ time.
- Identifying all edges takes $\Theta(?)$ time.

```
1  2  3  4  5  6  7  8
1 1 1 0 0 0 0 0 0
2 1 0 1 1 0 0 0 0
3 1 1 0 0 1 0 1 1
4 0 1 0 1 1 0 0 0
5 0 1 1 0 1 0 0 0
6 0 0 0 1 0 0 0 0
7 0 0 1 0 0 0 1 0
8 0 0 1 0 0 0 1 0
```

Graph Representation: Adjacency List

- Adjacency list. Node indexed array of lists.
- Two representations of each edge.
- Space proportional to _______.
- Checking if (u, v) is an edge takes $O(?)$ time.
- Identifying all edges takes $\Theta(?)$ time.

Connectivity

- s-t connectivity problem. Given two node s and t, is there a path between s and t?
- s-t shortest path problem. Given two node s and t, what is the length of the shortest path between s and t?

- Applications.
  - Friendster.
  - Maze traversal.
  - Kevin Bacon number.
  - Fewest number of hops in a communication network.
Breadth First Search

- BFS intuition. Explore outward from \( s \) in all possible directions, adding nodes one "layer" at a time.

- BFS algorithm.
  - \( L_0 = \{ s \} \).
  - \( L_1 = \) all neighbors of \( L_0 \).
  - \( L_2 = \) all nodes that do not belong to \( L_0 \) or \( L_1 \), and that have an edge to a node in \( L_1 \).
  - \( L_{i+1} = \) all nodes that do not belong to an earlier layer, and that have an edge to a node in \( L_i \).

For each \( i \), \( L_i \) consists of all nodes at distance exactly \( i \) from \( s \). There is a path from \( s \) to \( t \) iff \( t \) appears in some layer.

Breadth First Search - Analysis

Let \( T \) be a BFS tree of \( G = (V, E) \), and let \((x, y)\) be an edge of \( G \). Then the level of \( x \) and \( y \) differ by at most 1.

Theorem. The above implementation of BFS runs in \( O(m + n) \) time if the graph is given by its adjacency representation.