CS109B Notes for Lecture 4/14/95

Floyd's Algorithm

- Computes shortest path length for all pairs of nodes.
- Assumes adjacency matrix representation.
- Takes $O(n^3)$ time.
- Keeps matrix of distances: $M[i][j]$ is the shortest distance found so far from node i to node j.
- Initially, $M[i][j]$ = arc label for $i \to j$. If no arc $i \to j$, then $M[i][j] = 0$ if $i = j$ and ∞ otherwise.

Pivoting

The basic idea in Floyd's algorithm is to *pivot* on some node v.

- For all nodes i and j, replace $M[i][j]$ by $M[i][v] + M[v][j]$ if the latter is smaller.
- Inductive assertion: At any round, M has the shortest paths that go through only nodes on which we have pivoted previously.
- At end, we have pivoted on all nodes, so we have the shortest of all paths.

Running Time

- Each round takes $O(n^2)$ time to compare and update n^2 matrix entries.
- We must pivot on all n nodes, so $O(n^3)$ total.

Complete Graphs

Every possible edge/arc is present.

- Notation: K_n = complete undirected graph of n nodes.
  - □ Has $\binom{n}{2}$ edges.
- Complete directed graph of n nodes has n^2 arcs.

Bipartite Graph

An undirected graph for which it is possible to divide the nodes into two groups so that all edges connect one node from each group.

- *Complete bipartite graph* $K_{i,j} = i$ nodes in one group, j nodes in the other, and ij intergroup edges.

Planar Graphs

An undirected graph is *planar* if its nodes can be placed in a plane so that the edges do not cross.

- It is permitted to bend the edges.
- Kuratowski's theorem: A graph is planar iff it has no "copy" of either K_5 or $K_{3,3}$.
  - □ A "copy" of K_5 is a selection of 5 nodes such that there are *paths* (not necessarily single edges) between each two nodes. A "copy" of $K_{3,3}$ is defined similarly: 6 nodes, paths between pairs of nodes that have an edge in $K_{3,3}$.

Class Problem

How can we tell if a graph is bipartite?

- Note that if we are given the two groups of nodes, we can easily check that there are no intragroup edges in $O(m)$ time, assuming adjacency lists.
- However, we are *not* told what the two groups are; we are asked if two groups exist such that all edges are intergroup.
- Nevertheless, there is an $O(m)$ algorithm to find these groups if they exist.

Another Class Problem

A *tripartite* graph is one whose nodes can be divided into three groups, so that no edge connects two nodes of the same group.

- How fast can we tell whether a graph is tripartite?