These are the slides for Nobember 18, 1997.

Remember that examples will be written down in class but aren't prepared in advance on the slides.

Representing Graphs

A graph is a collection of combinatorial data. What's important is how the vertices and edges fit together, not any particular depiction of the edges and vertices on a piece of paper (or acetate).

One unimaginative way to represent a simple graph is to list each vertex, and for each vertex the set of vertices that it is connected to. For example, for the cycle C_4, one could write:

Vertex	Connected to
1	2 4
2	1 3
3	2 4
4	1 3

A better way, which we have already seen, is to set down an $n \times n$ matrix, where n is the number of vertices. (Order the vertices.) You write a "1" in the ijth place if the ith vertex is connected to the jth vertex and a 0 otherwise. The resulting matrix (for a simple graph) is symmetric and has 0s on the diagonal. In the previous example we would have the matrix

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

As we have already discussed in class, you can use these *adjacency matrices* to represent more complicated graphs. Undirected graphs are represented by symmetric matrices. When loops are present, the 0s on the diagonal can become 1s. When there are multiple edges between two vertices, the entries of the matrix become natural numbers that are not necessarily 0 or 1: in the

ijth place you put in the number of edges between vertex i and vertex j. Here's an example:

Similarly, we can deal with directed graphs. An edge coming out of the ith vertex and into the jth vertex contributes "1" to the entry in the ijth place. Here's a random example:

Incidence Matrices

The second way to encode a graph with a matrix is to have a matrix whose rows are in 1-1 correspondence with the vertices of the graph and whose columns are in 1-1 correspondence with the edges of the graph. You put a 1 in the ijth place if the jth edge touches the ith vertex. This works fine if there are multiple edges between vertices (add more columns). For loops, use only a single 1 in the appropriate row. This method pertains principally to undirected graphs, but you could invent a modification that applies to directed graphs as well. (Use -1s for tails and $+1$s for heads.) Here's an example:

Isomorphism

Two graphs are said to be isomorphic if they are "the same," except that vertices (and edges) might be labeled differently. If you suspect that two graphs are isomorphic, you can usually figure out what relabeling is necessary to make them the same. If they're not isomorphic and you want to prove this, you have to show that there is *no* possible relabeling which makes the graph the same. That's harder.

If graph$_1$ has some property that graph$_2$ doesn't, then the two graphs can't be the same. For example, if one graph is bipartite and the other isn't, then the two graphs are not isomorphic.

Connectivity

A path of length n from a vertex u to a vertex v is a sequence of n edges which begins at u and ends at v. (For a more formal definition, see the book.) The sequence of vertices that you get looks something like: u, x_1, x_2,..., x_{n-1}, v. The intermediate vertices x_1, ..., x_{n-1} are the vertices that the path "passes through."

A *circuit* is a path from u to u for some u. (It's a path that begins and ends in the same place.)

This concept pertains initially to undirected graphs, but you can imagine what the analogue is for directed graphs.

A funny example

Take a graph with two vertices A and B and two edges: an edge connecting A to B and a loop at A. How many paths of length n are there from A to B?

There's one path of length 1 (go from A to B). There is one of length 2 (do the loop, then go to B). There are two of length 3: you can do the loop twice and then go to B or go to B, come back to A and then return to B.

How many of length 4? You can do a loop and then execute a path of length 3 or you can go to B, come back to A and then execute a path (actually, *the* path) of length 2. So the answer is "3." Got the picture?

Some definitions and a theorem

An undirected graph is called *connected* if there's a path between every distinct pair of vertices.

A *simple* path is a path that never contains the same edge twice.

Theorem 1 *In a connected undirected graph, there is a simple path between every pair of distinct vertices.*

The statement is perhaps not completely obvious. A path from u to v might employ the same edge twice at different points. However, a path of smallest possible length from u to v will never pass through the same vertex twice....

A comment about isomorphism

A good way to show that two graphs are *not* isomorphic is to exhibit some path property in one graph that is absent in the other. For example, Rosen discusses two graphs with similar properties (shown on page 467). One has a simple circuit of length 3 but the other doesn't.

Adjacency matrices again

Suppose that G has adjacency matrix A. To find the number of paths of length n from vertex i to vertex j, look at the ijth entry of A^n. In the "funny example" before, $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$. If you calculate a few powers of A, you sense that $A^n = \begin{pmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{pmatrix}$, where the fs are the Fibonacci numbers. Can you prove this by induction?