MATHEMATICS S-152, SUMMER 2005
THE MATHEMATICS OF SYMMETRY

Outline #11 (Graphs)

References

This material is based on Biggs' Chapters 15 and 16 and on excerpts from Grossman and Magnus, *Groups and Their Graphs*. There are also a few pages of notes that were originally prepared for the Summer School course Computer Science S–111. Feel free to ignore the C code in the notes, which is taken from the program miles.exe (on the course Web site and distributed by email.) Some of these presentations are quite short.

For many of the proofs requested below, the only thing that is hard is to figure out what needs to be proved! Ask yourself what could go wrong, and devise a proof that rules it out.

The program miles.exe comes with lots of data files. If you want to run it on your Windows computer, download miles.zip from the course Web site and unzip everything into a single directory.

1. Present Biggs' definition of a graph (p. 178). Invent a graph with no more than 5 vertices and no more than 6 edges and represent it by a list of sets, a diagram, and an adjacency list. State how the preceding definition might prove inadequate for the following cases and suggest how it could be modified to cover them.

    (a) "Multigraph:" Vertices represent islands, and edges represent bridges between them. There may be two bridges between the same pair of islands.

    (b) "Directed graph: "Vertices represent elements of a group, and the edge joining a to b represents the element g such that $ga = b$.

    (c) "Graph with self loops:" Vertices represent campgrounds in a national park, and edges represent trails that connect them to one another. Then someone builds a "loop nature trail" that starts and ends in the same campground.

2. Define the complete graph K_n (Biggs, p. 179, exercise 3). Draw pictorial representations for K_3 and for K_4 in which edges do not cross. Show that this cannot be done for K_5, even if the rules are relaxed to allow edges to be represented by curves instead of straight line segments.

3. Define what is meant by isomorphism of graphs (Biggs, section 15.2). Invent an example of two graphs that are isomorphic (even though it may not be apparent from their pictorial representations) and of two graphs that have equal numbers of vertices and edges but that are not isomorphic.

4. Define the degree of a vertex of a graph, and prove that the number of odd vertices is even (Biggs, p. 182).

5. Define the terms "walk," "path," and "cycle" for a graph (Biggs, p. 183 and 184). Define "Hamiltonian cycle" and "Eulerian walk." State and prove a necessary condition for a connected graph to have an Eulerian walk and to have an Eulerian walk which starts and ends at the same vertex (often called an "Eulerian cycle," although it is not necessarily a cycle by Biggs' definition).

6. (a) Show that for the bridges of Koenigsberg (last page of the attached notes) there is no Eulerian walk. This was the problem that inspired Leonhard Euler, the greatest mathematician of the eighteenth century, to invent the branch of mathematics now known as topology.
   (b) Draw a diagram of the complete graph K_5 (Biggs, p. 179, exercise 3), which has five vertices each with degree 4, and show an example an Eulerian walk.
   (c) Show that the complete graph K_5 has two Hamiltonian cycles that have no edge in common.

7. Define a tree T as a connected graph with no cycles (Biggs, p. 185). Prove the following properties of trees:
   (a) for each pair of vertices x and y, there is a unique path in T joining x and y.
   (b) the graph obtained from T by removing an edge has two components, each a tree.

(c) $|E| = |V| - 1$.

8. Define "minimum spanning tree" and state and prove the minimum spanning tree property (page 10–18 of the notes). (This is the first part of the proof of Theorem 16.3 in Biggs, but it makes sense to do the proof for an arbitrary set S of vertices.)

9. Describe Prim's algorithm for constructing a minimum spanning tree. Illustrate its action on the graph shown on page 10–19 of the notes, and prove that it is correct. The proof is easy, since the minimum-spanning-tree property has already been proved. (This algorithm, for a non-weighted graph, is described at the start of Biggs, section 16.3., and for a weighted graph it is the "greedy algorithm" that Biggs describes on pp. 200–201.)

10. Describe Kruskal's algorithm for constructing a minimum spanning tree. Illustrate its action on the graph shown on page 10–21 of the notes, and prove that it is correct. The proof is easy, since the minimum-spanning-tree property has already been proved. (This algorithm is not mentioned in Biggs.)