

# **Applications of Elementary Graph Theory**

**Charles Mullins**  
**Math Dept, ASMS**  
**[mullinsc@asms1x.dsc.k12.ar.us](mailto:mullinsc@asms1x.dsc.k12.ar.us)**  
**May 2002**

## Applications of Elementary Graph Theory

Charles Mullins

Math Dept, ASMS

[mullinsc@asms1x.dsc.k12.ar.us](mailto:mullinsc@asms1x.dsc.k12.ar.us)

### Objectives:

- 1) Students will see an application of graph theory to a classic problem known as the Königsburg Bridge Problem and be able to apply results on Euler Paths and Circuits to solve related problems.
- 2) Students will know what the Traveling Salesman Problem is and discover how difficult it can be for even a relatively small number of cities. Students will be able to apply the Nearest Neighbor Algorithm to such problems

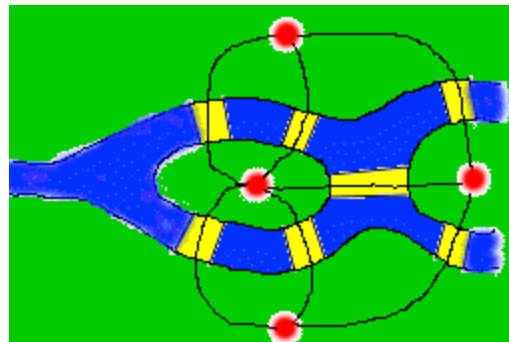
### Key Points:

- 1) A graph consists of points and edges. An edge is an arc (or line segment) joining two points. Points are also called vertices. A path joins one point (vertex) successively to a sequence of other points (vertices) along edges. A circuit is a path that starts and ends at the same point. An Euler path or circuit passes through every vertex exactly one time.
- 2) The Traveling Salesman Problem is the problem of visiting a collection of cities in the most efficient manner so that no city is visited twice. The Nearest Neighbor Algorithm is a quick and efficient method of approaching the Traveling Salesman Problem although it usually does not give the best solution.

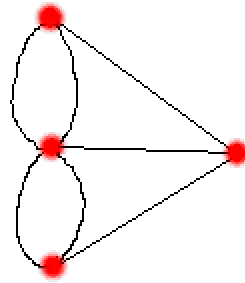
### **The Bridges of Königsburg. (1 Day)**

One of the classic problems in Graph Theory is the Bridges of Königsburg. Königsburg is a city in Germany with seven bridges joining various parts of the city across the Pregel River (see figure below).

To mathematicians, though, Königsburg is best known because of a puzzle associated with its seven bridges, which were located roughly as illustrated on the right. Its citizens pondered for a long time whether it was possible to walk about the city in such a way that you cross all seven bridges (yellow in diagram) exactly once.



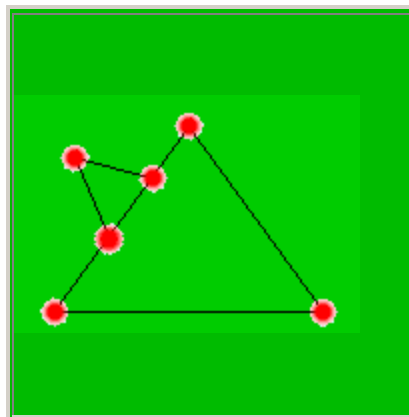
As mentioned above, on a pleasant afternoon, a favorite pastime of people in Konigsburg was to try and walk a path across each bridge and return to their starting point crossing each bridge exactly one time. The great mathematician Euler (pronounced “oiler”) realized that this task could be treated a problem in Graph Theory by stripping it down to the bare essentials. The various parts of the city separated by the river could be treated as points (called *vertices* in Graph Theory) and the paths between them across the bridges as line segments or arcs (called *edges*). Below is this stripped down version of the seven bridges problem.



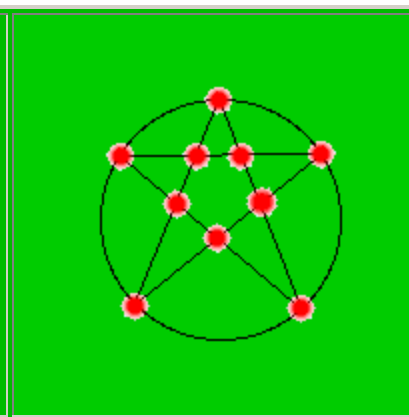
Using this stripped down version of the problem, Euler was able to give an argument that showed that such a circuit did not exist; that is, he proved that it was impossible to make such a walk. This idea has been extended to arbitrary graphs and a path starting at a certain vertex and passing along each edge (across each bridge) exactly one time ending at the original vertex is called an *Euler Circuit* in honor of the man himself.

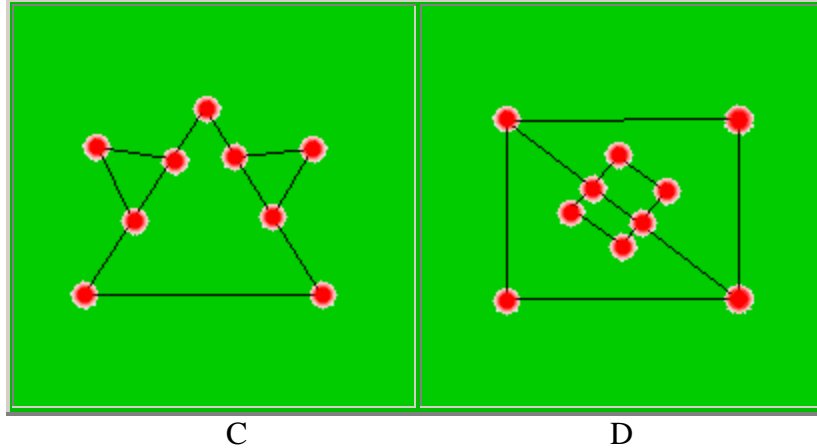
- 1) At this point students should try to find an Euler Circuit for the Konigsburg bridges.
- 2) After many attempts, ask students if they believe it is possible and why or why not. Inform the students that there is no Euler Circuit but before you explain why have them try and find Euler Circuits for the graphs A, B, C, and D below.

A



B





It may be possible at this point for some students to come up with a argument similar to the one Euler gave. Euler realized that unless one is at the starting vertex, for any Euler Circuit, every time the path goes into a vertex it must also come out or else one would be stuck at that vertex since you are not allowed to retrace any edge. Hence, except for the starting vertex, each vertex must have an even number of edges attached to it (such a vertex is said to have *even degree*). As for the starting vertex it must also have even degree since you start out from there and every time you come back to it (if any) you will go out until you finish there. Heuristically then it appears that ALL vertices must have even degree. Since NONE of the vertices in the Konigsburg bridge graph have even degree, there certainly cannot be an Euler Circuit. In fact, Euler gave a rigorous proof of the following theorem on the existence of an Euler Circuit.

Theorem: A graph has an Euler Circuit if and only if it is connected and all its vertices are of odd degree.

Note: *Connected* means that every vertex is joined to at least one other vertex.

3) Have students use the theorem to check their answers about the graphs A,B,C, and D above that have Euler Circuits.

Again, a path that begins and ends at the same vertex is called a circuit and if it covers each edge exactly once, it is called Eulerian or an Euler Circuit. Then, a sleight variation on the problem is to allow the path to end at a different vertex than the starting one, that is, the Euler Path does not have to be a circuit but does have to cover every edge exactly once. In this case, the above theorem requires a slight change.

4) Ask students how this will change the Theorem.

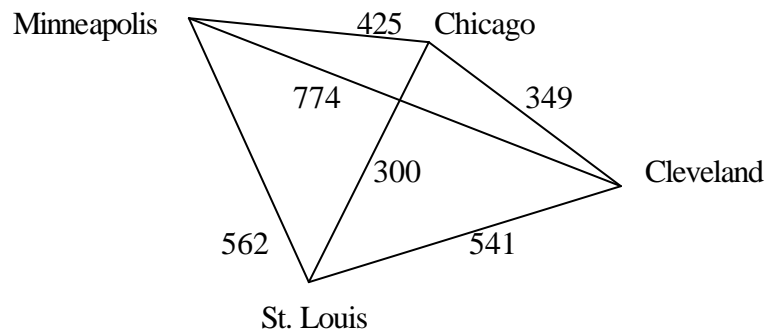
5) To evaluate if the students have grasped these ideas, have them determine which of the graphs above have Euler Paths. Having explored these ideas, they should also be able to find Euler Paths for those graphs.

All the graphs above were taken from the website on the Konigsburg Bridge Problem, Math Dept, The National University of Ireland, Maynooth:  
<http://www.maths.may.ie/images/konig.html>

### The Traveling Salesman Problem (TSP). (2 Days)

A Graph Theory problem that has many applications and appears to be very realistic is the Traveling Salesman Problem: Given a set of cities (or locations) and the distances between each pair of cities (or locations), find a path which visits each city exactly one time such that the total distance traveled is minimum. This problem lends itself readily to Graph Theory as the locations can be represented by vertices and, for those cities which are joined by some edge, the edge can be labeled with the distance (in real life it may not be possible to travel from a location to all other locations-the version of TSP here only considers the case where each location is indeed connected to all the others).

1) Have students solve the TSP for the graph below, i.e. find a path that passes through each city for which the total distance (in miles) is minimum. For convenience, it is best to specify which vertex is the starting point although this requirement will not affect the optimal solution.



Students will solve the problem by trying all examples and picking the one that gives the minimum value. This technique is known in Graph Theory as *Brute Force*. It is the only known method for ensuring that one has actually found the path that gives the minimum value. For small graphs it is a reasonable approach and can be programmed for calculators and computers. However, it doesn't take many vertices to produce a situation where the time needed for this algorithm becomes quite large. In the case of 4 cities starting at some city, there are 3 choices for the next leg of the journey and then 2 for the next leg and finally 1 choice for the final leg. The Fundamental Theorem of Counting then gives the total number of paths as  $3 \times 2 \times 1 = 6$ . Students should have also seen that there is a symmetry among all the possibilities, i.e. the path St. Louis-Cleveland-Chicago-Minneapolis-St. Louis will produce the same mileage as its reverse, St. Louis-Minneapolis-Chicago-Cleveland-St. Louis. Hence, instead of 6 possibilities there are really only  $6/2 = 3$  "distinct" ones. In general, if there are  $n+1$  cities for some positive integer  $n$ , then, starting at some city, there are  $n$  choices for the first leg,  $n-1$  for the

second leg,  $n-2$  for the third leg, until we get down to 3 choices, then 2, then 1 choice. Again using the Fundamental Theorem of Counting the total number of paths is  $n(n-1)(n-2)\dots 3*2*1$  which is  $n$  factorial and is denoted by  $n!$ . As in the case of 4 paths above, due to symmetry the result  $n!$  must be divided by 2 to obtain the number of “distinct” paths.

2) Students should calculate:

a)  $4!$     b)  $5!$     c)  $10!$     d)  $20!$

Look at how large  $20!$  is. And this is “only” twenty-one cities. How about  $100!$  (a hundred and one cities)?

3) If a really good computer can perform one billion ( $10^9$ ) operation per second, it could handle the expression  $100!$  in a) \_\_\_\_\_ seconds. Is this a reasonable number? To see what this number really means have students convert units.

- b) \_\_\_\_\_ minutes
- c) \_\_\_\_\_ hours
- d) \_\_\_\_\_ days
- e) \_\_\_\_\_ weeks
- f) \_\_\_\_\_ months
- g) \_\_\_\_\_ years
- h) \_\_\_\_\_ centuries
- i) \_\_\_\_\_ millennia
- j) \_\_\_\_\_ ??????(can they go further?)

It should be clear at this point that some alternative to the Brute Force method for solving the TSP is worth exploring. Unfortunately, no totally satisfactory method has been obtained, that is, there is no known algorithm (step-by-step procedure) for obtaining a path that gives the minimum. Moreover, some researches believe that no such algorithm will ever be found. To explore the TSP further, students can go to the TSP Home Page: <http://www.Princeton.edu/tsp>. This site gives a picture of the solution for 15,112 cities-remember there are  $15,111!/2$  “distinct” paths altogether.

Rather than insist on an algorithm that guarantees the best solution i.e. one that gives the minimum, researchers have developed algorithms that give good results much of the time and seem reasonable. Such algorithms are sometimes called heuristic because they are quick and seem reasonable even if they may not always produce the best result. One of these is called the *Nearest Neighbor Algorithm*:

*From the starting city pick the nearest city, and then at each stage thereafter, go to the nearest city that has not already been visited (if there are several unvisited cities the same distance away, choose any of them).*

Hence, in the original problem of four cities above, starting at Chicago, first travel 300 miles to St Louis, then 541 miles to Cleveland. Since we must visit Minneapolis, we then travel the 774 miles to it from Cleveland and then the 425 miles back to Chicago. The total mileage is  $300+541+774+425=2040$  miles. Note that this does not give the minimum distance but it has the great advantage that it is easy to use no matter how many cities are involved and it gives a “reasonable” answer.

4) Have students construct the graph and use the Nearest Neighbor Algorithm for the following situation below starting and ending at A. Can they find a path that gives a smaller value? Repeat starting at other sites.

	A	B	C	D	E
A		40	55	85	80
B	40		75	95	70
C	55	75		65	60
D	85	95	65		90
E	80	70	60	90	

5) Have students create their own graph with their favorite cities. Then, apply the Nearest Neighbor Algorithm to it. To use the Nearest Neighbor Algorithm, students can have a lot of cities but if they wish to compare it to the best solution found by *Brute Force*, they will need to keep the number of cities below 6-remember for 6 cities there are  $5!/2=60$  “distinct” paths, whereas for 5 cities, there are only  $4!/2=12$  “distinct” paths. Does the choice of starting point affect the result? Does the Nearest Neighbor Algorithm give the minimum?

Revised, 16 May 2002

## ANSWERS

### PAGE 3

1) The Konigsburg bridge graph has no Euler circuit. As we shall see later this is due to the fact that it has a vertex with an odd degree (in fact, ALL vertices have odd degree).

2) Only B has an Euler Circuit. Again, all other graphs have a vertex of odd degree.

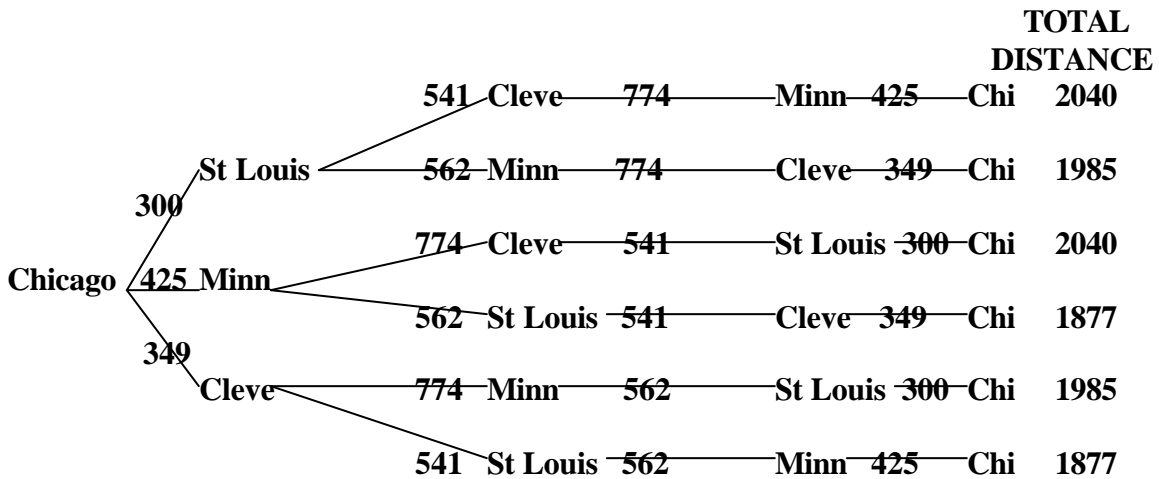
### PAGE 4

4) A graph will have an Euler Path which is not an Euler Circuit, if it has exactly 2 vertices of odd degree. The Eulerian Path must begin at one of the vertices of odd degree and end at the other one.

5) Graphs A and D have Euler Paths since they have exactly 2 vertices of odd degree.

### PAGE 5

1) We start at Chicago but it doesn't matter as these are circuits so the total distance doesn't depend on the starting point.



Using a tree diagram all possible routes are constructed with the total mileage (the method of *Brute Force*). Clearly the shortest distance is 1877 miles and occurs for Chicago-Minneapolis-St. Louis-Cleveland-Chicago and Chicago-Cleveland-St. Louis-Minneapolis-Chicago which is just the first one traveled in the opposite direction (so there are really only 3 “distinct” possibilities).

PAGE 6

2) a) 24 b) 120 c) 3,628,800 d)  $2.43290 \times 10^{18}$

3) a)  $9.33262154 \times 10^{157}$

The TI89 can actually compute 100! but the TI83 gives an overflow error. There are many ways around this problem. One is to compute  $60! = 8.320987113 \times 10^{81}$  (actually the limit of the TI83 is 69!) and then do  $100 \cdot 99 \cdot 98 \cdot \dots \cdot 62 \cdot 61 = {}_{100}P_{40} = 1.121576253 \times 10^{76}$ . Multiplying  $8.320987113 \cdot 1.121576253$  and rounding to 8 places yields the desired result above.

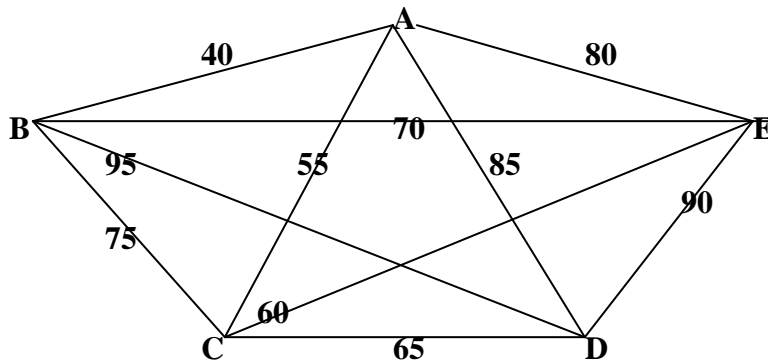
b)  $1.555436923 \times 10^{147}$  c)  $2.592394872 \times 10^{145}$  d)  $1.08016453 \times 10^{144}$

e)  $1.543092186 \times 10^{143}$  f)  $5.14364062 \times 10^{141}$  (using 30 days in a month)

g)  $4.286367183 \times 10^{140}$  h)  $4.286367183 \times 10^{138}$  i)  $4.286367183 \times 10^{137}$

PAGE 7

4)



Starting at A: A—40—B---70---E---60---C---65---D---85---A for a total of 320

Starting at B: B---40---A---55---C---60---E---90---D---95---B for a total of 340

Starting at C: C---55---A---40---B---70---E---90---D---65---C for a total of 320

Starting at D: D---65---C---55---A---40---B---70---E---90---D for a total of 320

Starting at E: E---60---C---55---A---40---B---95---D---90---E for a total of 340

Clearly, in the case of the Nearest Neighbor Algorithm, the starting point makes a difference but the best value is 320.

Note that the circuit C-A-B-E-D-C is the “same” as D-C-A-B-E-D.

Brute Force shows that the best possible value is also 320.

*Note: While I have tried to check all the answers above, errors can occur. If you discover any, I would greatly appreciate it if you would notify me.*

