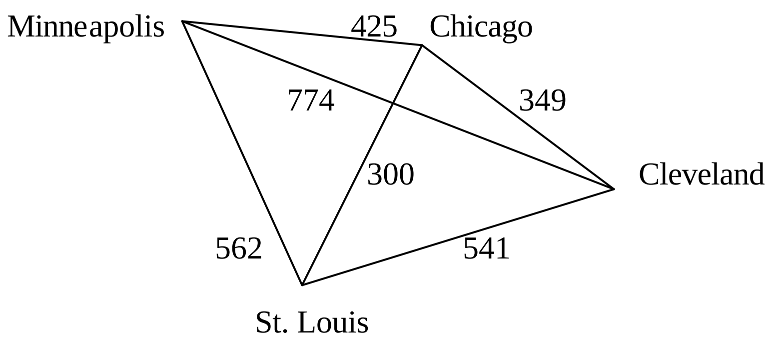
The Traveling Salesman Problem (TSP). (2 Days)

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 choice 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*2*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)…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?