Hour-Exam on Friday, June 21

- Covers material in Chapters 1, 2, 3
- Bring #2 pencil and picture ID.
- You may use a calculator.
- You may NOT use cell phones or other wireless devices.
- You may NOT use books or notes.
- There will be room on the exam paper for calculations.

HAMILTONIAN CIRCUITS

A Hamiltonian circuit is a circuit that includes each vertex of the graph once and only once. (At the end, of course, the circuit must return to the starting vertex.)

Traveling Salesman Problem

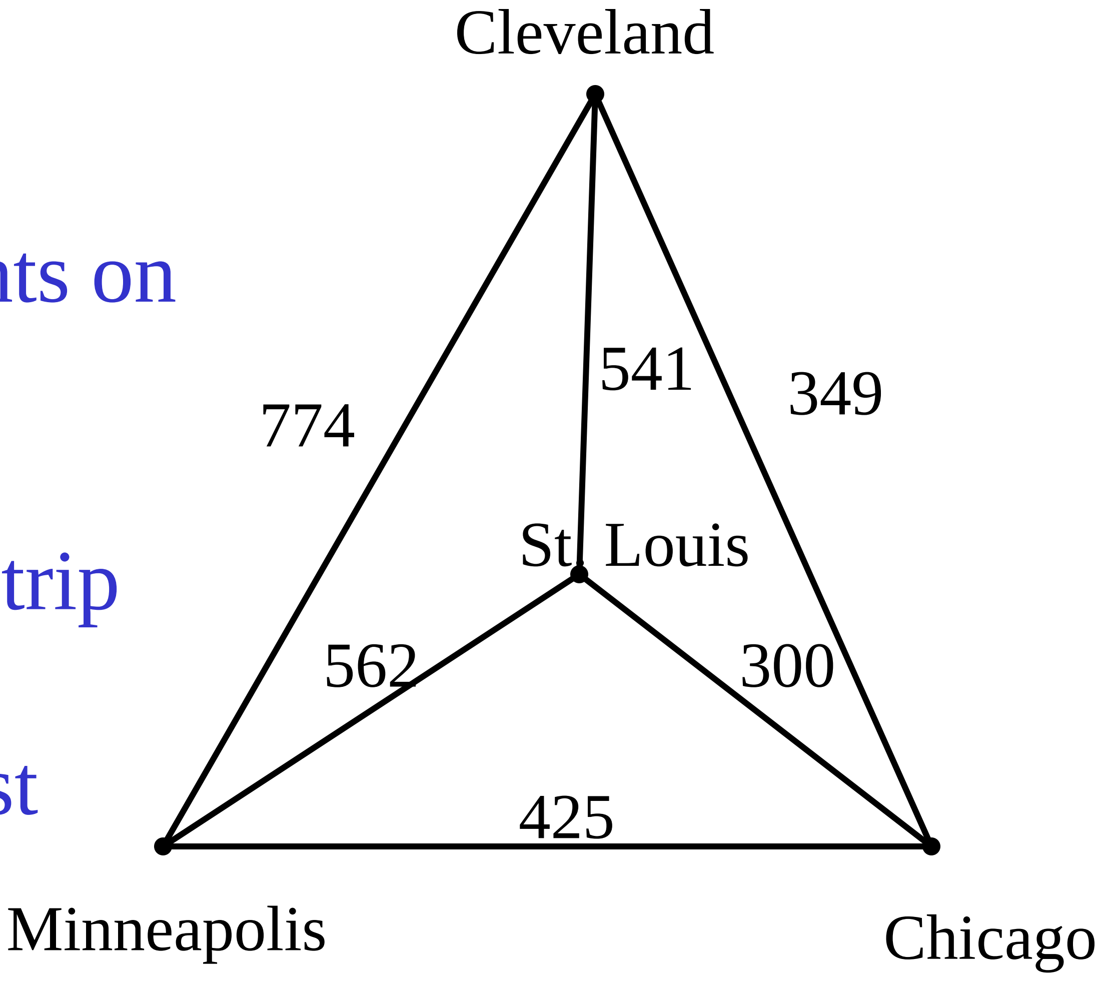
Finding a minimum-cost Hamiltonian circuit in a complete graph with *N* vertices is usually called the traveling salesman problem (TSP).

A complete graph with *N* vertices has $(N - 1)!$ distinct Hamiltonian circuits.

$$(N-1)! = (N-1)(N-2)\cdots 3\cdot 2\cdot 1$$

Planning a trip

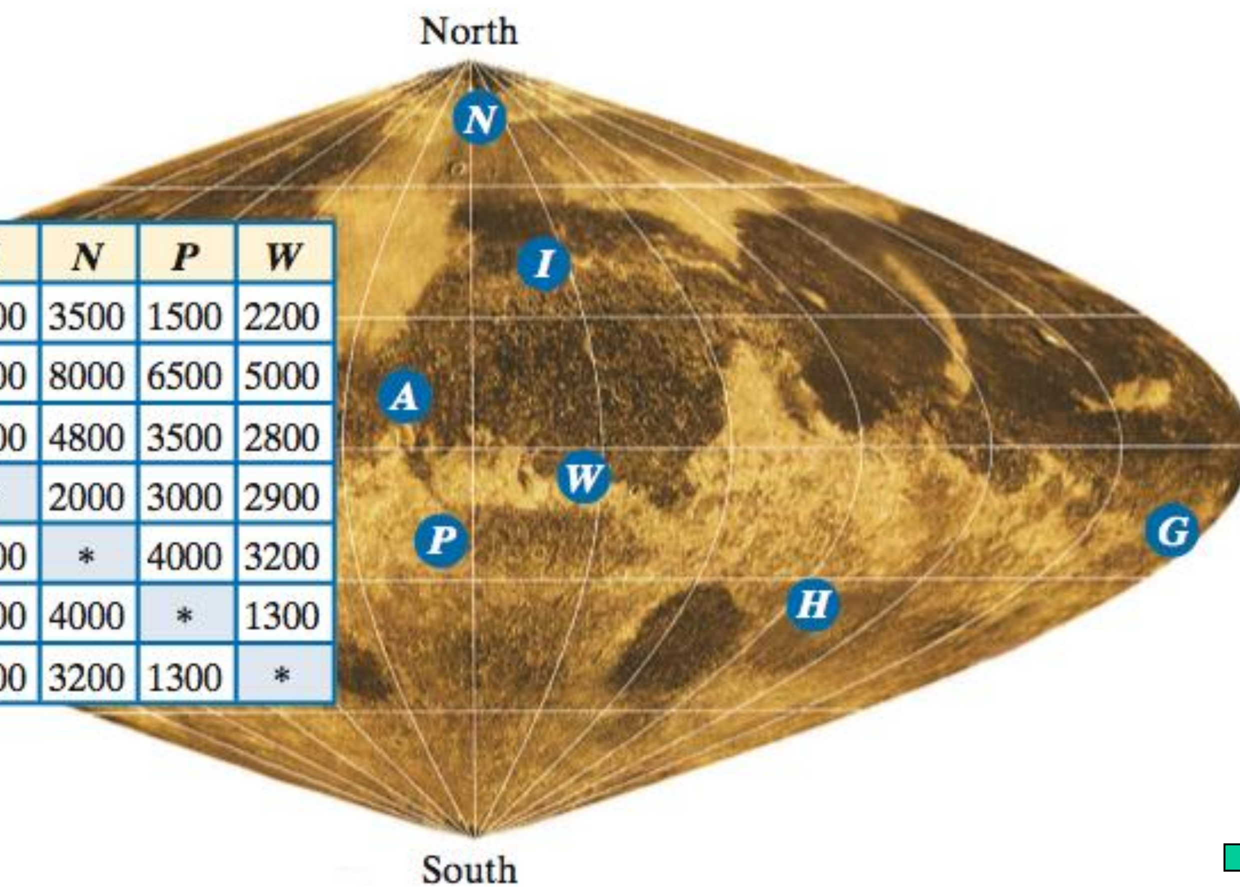
This problem uses the complete weighted graph with four vertices. The weights on the edges are the distances between cities. The optimal trip is the Hamiltonian circuit with the least total weight.



Example 2.6 Roving the Red Planet

Here are the estimated distances (in miles) that a rover would have to travel to get from one Martian site to another. What is the optimal (shortest) tour for the Mars rover?

	A	G	H	I	N	P	W
A	*	7500	5000	2800	3500	1500	2200
G	7500	*	3000	6000	8000	6500	5000
H	5000	3000	*	4000	4800	3500	2800
I	2800	6000	4000	*	2000	3000	2900
N	3500	8000	4800	2000	*	4000	3200
P	1500	6500	3500	3000	4000	*	1300
W	2200	5000	2800	2900	3200	1300	*



ALGORITHM 1: THE BRUTE FORCE ALGORITHM

Step 1. Make a list of *all* the possible Hamilton circuits of the graph. Each of these circuits represents a tour of the vertices of the graph.

Step 2. For each tour calculate its weight (i.e., add the weights of all the edges in the circuit).

Step 3. Choose an *optimal* tour (there is always more than one optimal tour to choose from!).

Enumeration is time-consuming!

If we have 25 cities, then there will be

$$24! \approx 6 \times 10^{23}$$

Hamiltonian circuits

If these Hamiltonian circuits are generated at the rate of 1 billion per second, it would take 20 million years to generate them all.

Number of Distinct Hamilton Circuits in K_N

N	$(N-1)!$	N	$(N-1)!$
3	2	12	39,916,800
4	6	13	479,001,600
5	24	14	6,227,020,800
6	120	15	87,178,291,200
7	720	16	1,307,674,368,000
8	5040	17	20,922,789,888,000
9	40,320	18	355,687,428,096,000
10	362,880	19	6,402,373,705,728,000
11	3,628,800	20	121,645,100,408,832,000

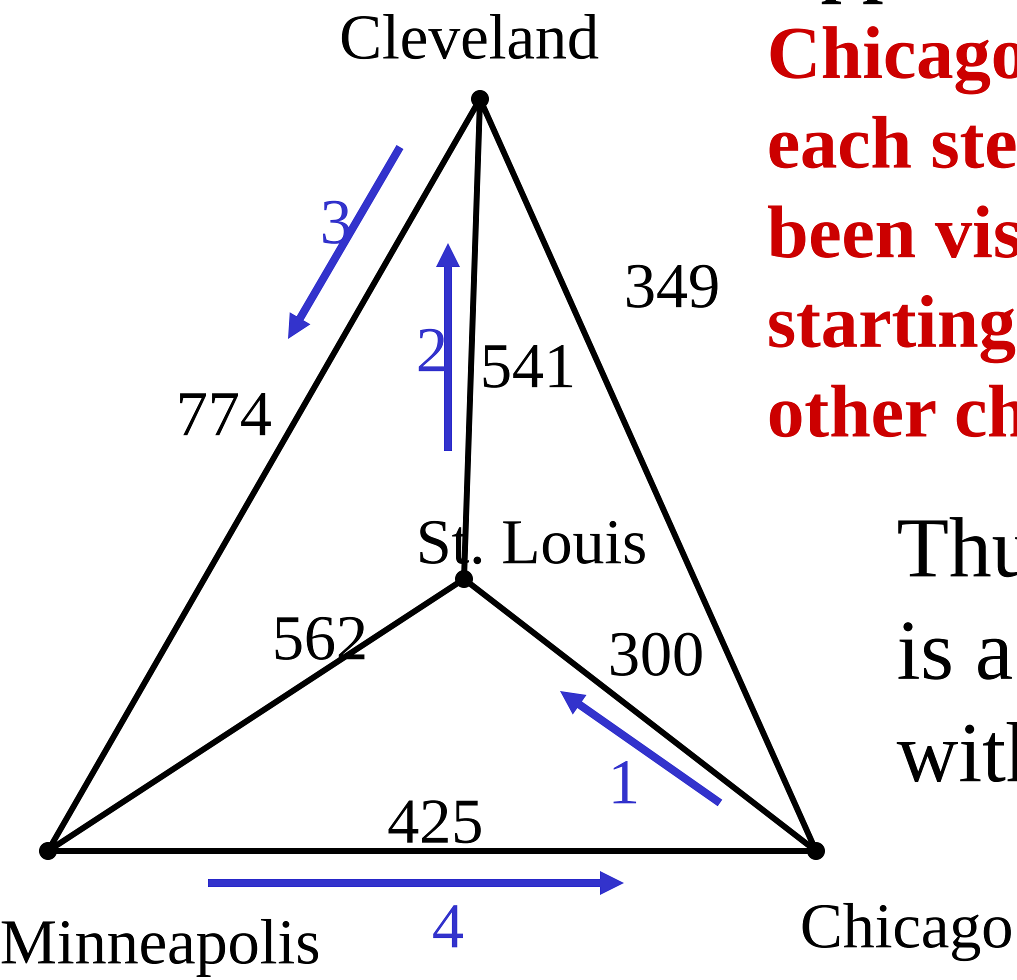
ALGORITHM 2: THE NEAREST NEIGHBOR ALGORITHM

Start: Start at the designated starting vertex.

Steps: From each vertex go to its nearest neighbor (i.e., follow the edge that has the smallest weight).

Finish: Return to the starting vertex when all other vertices have been visited.

Nearest-Neighbor Algorithm



Approach: starting from Chicago, visit the nearest city at each step that has not already been visited. We return to the starting city when there is no other choice left.

Thus: C, S, CL, M, C
is a Hamiltonian circuit
with total distance = 2040.

Pros and Cons

- Efficient: We only have to examine each of the $N(N-1)/2$ edges once to find the nearest neighbors and calculate the total for only one circuit.
- Not necessarily optimal: In this problem, this method produced the longest tour. The **relative error** in this method is:

$$\frac{(\text{cost of tour} - \text{cost of optimal tour})}{\text{cost of optimal tour}} = \frac{2040 - 1877}{1877} \approx 8.7\%$$

ALGORITHM: THE CHEAPEST LINK ALGORITHM

Steps. At each stage, pick the *cheapest link* (i.e., the edge with smallest weight) available.

Rule 1. Do not close a circuit until all vertices have been visited.

Rule 2. Do not create three edges coming out of a single vertex.

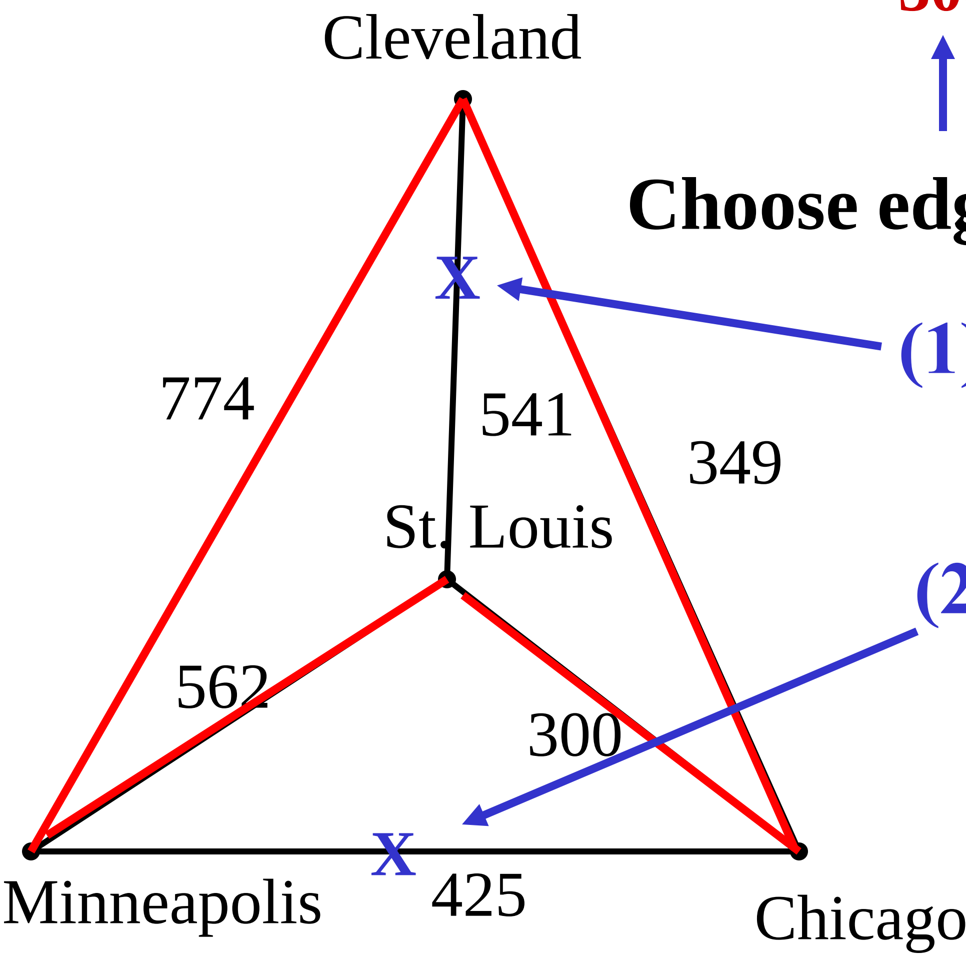
Put the six weights on the edges in increasing order:

300, 349, 425, 541, 562, 774

Choose edges in order so that we:

(1) don’t close the circuit too soon

(2) don’t create three edges coming out of a single vertex.



Solution By Cheapest-Link Algorithm

C, S, M, CL, C
Total = 1985

It is still not the optimal solution, but it provides us a quick way to find a Hamiltonian circuit. In this example, the Cheapest-Link Algorithm gives a little better result than the Nearest-Neighbor Algorithm, but for some problems the reverse is true.

Example 2.12 Roving the Red Planet

What is the tour if we use the Cheapest-Link Algorithm to find a Hamiltonian circuit that visits each of these sites?

Total length: 21,400 miles

	A	G	H	I	N	P	W
A	*	7500	5000	2800	3500	1500	~~2200~~
G	7500	*	3000	6000	8000	6500	5000
H	5000	3000	*	4000	4800	3500	2800
I	2800	6000	4000	*	2000	~~3000~~	~~2900~~
N	3500	8000	4800	2000	*	4000	3200
P	1500	6500	3500	3000	4000	*	1300
W	2200	5000	2800	2900	3200	1300	*