**Notes for 11/22/2010:**

**Graph Theory:**

**.** Node/ Vertex

\_\_\_\_\_\_ Edge connecting nodes:

1. Directed Edge, with arrow 🡪, like one-way road.
2. Undirected Edge, without arrow, like two-way road.

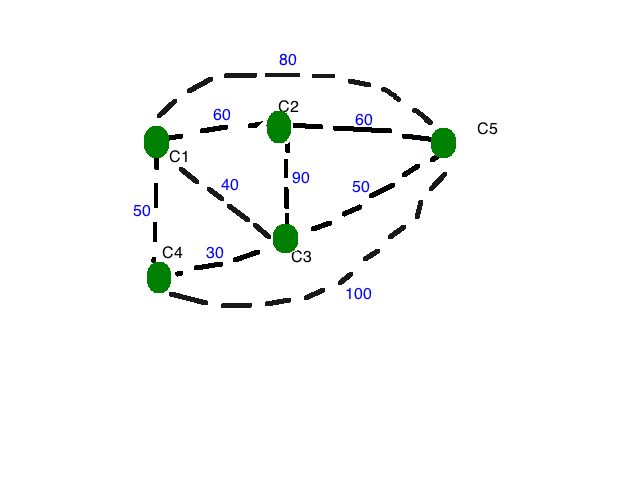
Hub: key point, disconnect a hub will make the network disconnected or will remove many possible paths.

**Concept 1: Minimum Spanning Tree:**

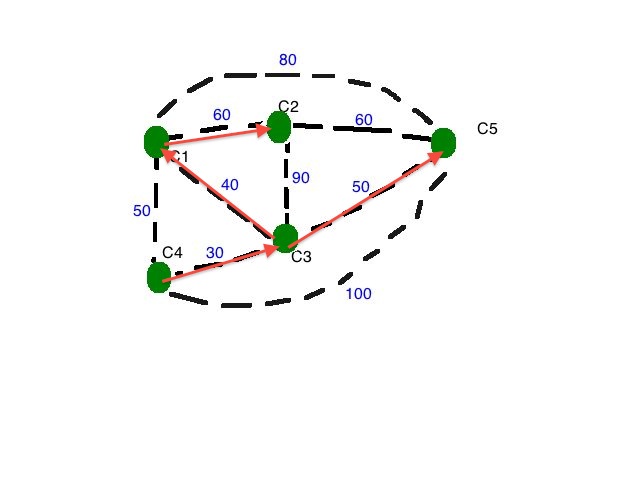
Using when choosing least-cost path only required basic connection.

Algorithm: Result={ }

Until every pair of nodes is connected (i.e. path from one to the other

Find the least-cost edge that connects two cities that are not already connected, then add to result.

Example: Try to connect five cities by building road with the least cost.

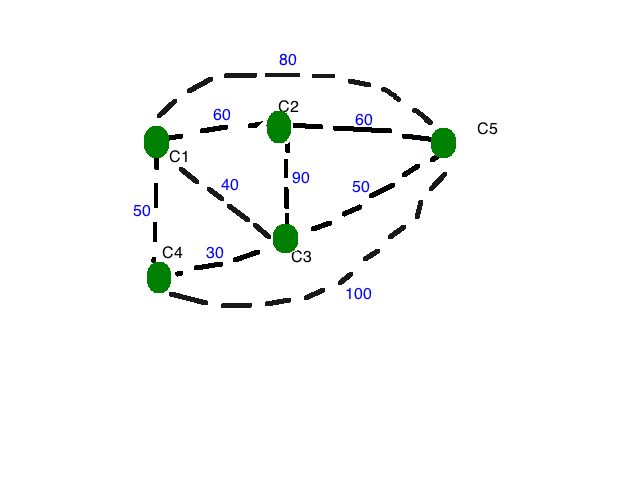
1. C4-C3
2. C1-C3
3. C3-C5
4. C1-C2/C2-C5

**Concept 2: Shortest Path Problem**

Strategy: Ask who’s my closest neighbor from start point

Algorithm: Frontier = { }

Until frontier includes destination.

Add to frontier Node V with cost C, such that V is not already in frontier and C is the cost from start point to V and for any other node V’, the cost from start point to V’ is greater than or equal to C.

Example 1:

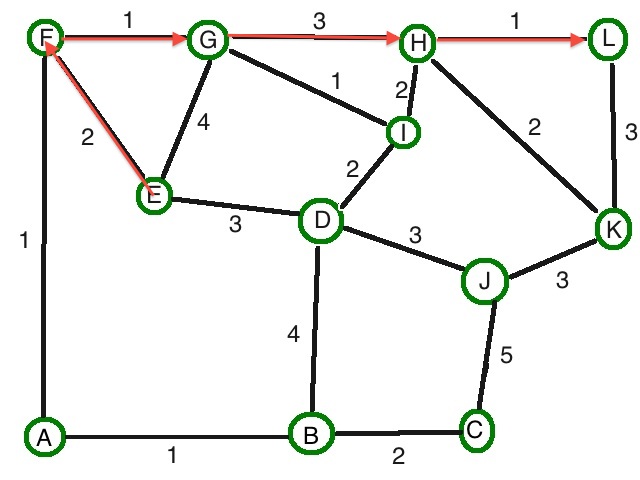
What’s the shortest path from C4 to C2?

Solution: {C4,0}🡪{C3,30}🡪{C1,50}🡪 {C5,80}🡪{C2,110}



Example 2:

Try to find the shortest path from E to L.

Solution:

{E,0}

🡪{F, 2, (E)} #cost=2, route passes E

🡪{A, 3, (F,E)} / {G, 3, (F,E)} / {D, 3 , (E)}

🡪{B, 4, (A,F,E)} / {I, 4, (G,F,E)}

🡪{H, 6, (G,F,E)/(I,G,F,E)} / {J, 6, (D,E)} /

{C, 6, (B,A,F,E)}

🡪{L, 7, {H,G,F,E)/(H,I,G,F,E)}

Remark: no negative cost.