MATH F113X: Sortest Edges/Cheapest Link Algorithm for Hamiltonian cycles

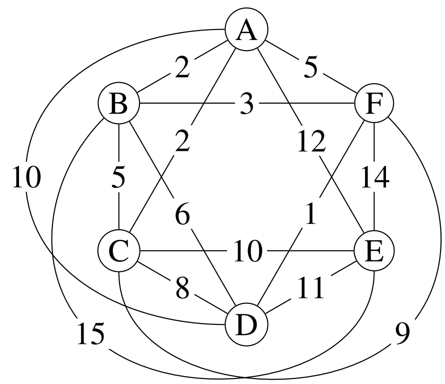
The Sorted Edges / Cheapest Link Algorithm

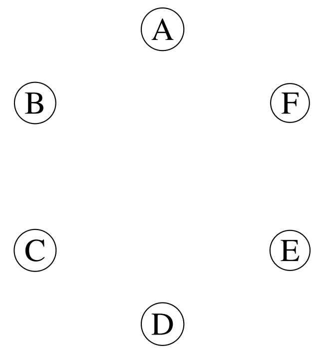
Steps: Add the next cheapest edge to your circuit **unless**

1. it closes the circuit too soon, or
2. creates a degree 3 vertex.

Break ties by choosing the alphabetically smallest edge.

Apply the Sorted Edges Algorithm to find a Hamiltonian circuit. Draw in the edges, labeled with their weight, as you add them on the empty graph.





Sorted edges	weight	used? (or why not)
FD	1	
AB	2	
AC	2	
BF	3	
AF	5	
BC	5	
BD	6	
CD	8	
CF	9	
AD	10	
CE	10	
DE	11	
AE	12	
EF	14	
BE	15	

List the vertices of the Hamiltonian circuit, starting at vertex A.

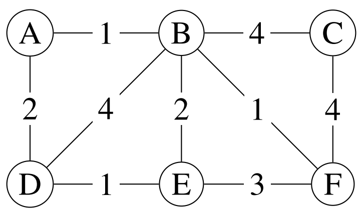
Total weight of the circuit?

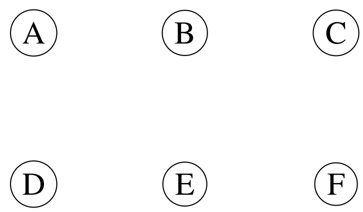
Do you think the circuit we obtained in the best possible?

MATH F113X: Sortest Edges/Cheapest Link Algorithm for Hamiltonian cycles

Can you construct a graph such that the Sorted Edges Algorithm will never result in a Hamiltonian circuit of smallest weight? What does this tell us about the Sorted Edges Algorithm?

What happens if you apply Sorted Edges/Cheapest Link to the following graph?





Sorted edges	weight	used? (or why not)
AB	1	
BF	1	
DE	1	
AD	2	
BE	2	

Sorted edges	weight	used? (or why not)
EF	3	
BC	4	
BD	4	
CF	4	

What is the problem here?

There is a Hamiltonian circuit on this graph. What is the smallest-weight Hamiltonian circuit you can find?

Circuit: ______________________________ Weight: __________