## MATH F113X: Sortest Edges/Chepaest Link Algorithm for Hamiltonian cycles

The Sorted Edges / Cheapest Link Algorithm

input: a graph with distances (weights) on the edges

**output:** a Hamiltonian circuit (or the algorithm fails)

**strategy:** Add the next cheapest edge to your circuit unless it closes the circuit too soon or creates a degree 3 vertex. Break ties by choosing the alphabetically smallest edge.

Note that you may have disconnected edges during the process of running the application.

For convenience, I have listed the edges of the graph in sorted order for you. Draw in the edges 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.

What was the weight of the circuit you found?

## MATH F113X: Sortest Edges/Chepaest Link Algorithm for Hamiltonian cycles

A

(

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

used? (or why not)



weight

1

1

1

1

1

1

2

2

Sorted edges

AB

BF

DE

DG

EG

HI

AD

BE

Œ	) E	F	)
G	H)	I	)
	Sorted edges	weight	used? (or why not)
	EH	2	
-	FI	2	
	GH	2	
-	BC	4	
-	BD	4	
-	$\overline{CF}$	4	

4

C

 $(\mathbf{B})$ 

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

EI



One strategy for having the algorithm not fail is to turn your graph into a complete graph, and assign very expensive weights (say, 100000) to the "non-edges". For example, there is no edge between G and A, so you could add edge AG and give it a weight of 10000.

The complete graph on 9 vertices has 36 edges.

- 1. How many edges would you need to add to turn this graph into a complete graph?
- 2. In this new graph, how many possible Hamiltonian circuits are there?