Algebra and Discrete Mathematics (ADM)

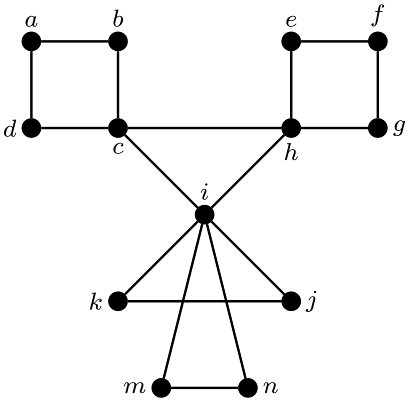
Tutorial 9 Hamiltonian cycles

Lecturer: Bc. Xiaolu Hou, PhD.
xiaolu.hou@stuba.sk

Necessary conditions for existence of Hamiltonian cycles

- G must be connected
- No vertex of G can have degree less than 2
- G cannot contain a cut-vertex

The last condition is not satisfied: i, h, c



Existence of Hamiltonian cycles

Necessary conditions:

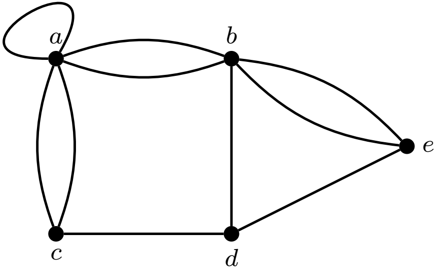
- G must be connected
- No vertex of G can have degree less than 2
- G cannot contain a cut-vertex

A sufficient condition (Dirac's Theorem):

- Number of vertices $n \geq 3$ vertices.
- Every vertex v of G satisfies $\deg(v) \geq \frac{n}{2}$

$$n = 5, \quad \frac{n}{2} = 2.5$$

One Hamiltonian cycle: $abedca$



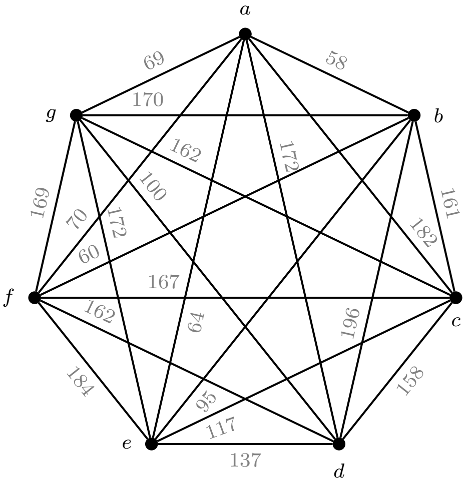
Nearest Neighbor Algorithm

1. Choose a starting vertex, say v. Highlight v
2. Among all edges incident to v, pick the one with the smallest weight. If more than one possible choices have the same weight, randomly pick one.
3. Highlight the edge and move to its other endpoint u. Highlight u
4. Repeat steps 2 and 3, where only edges to unhighlighted vertices are considered
5. Close the cycle by adding the edge to v from the last vertex highlighted. Calculate the total weight.

Nearest Neighbor Algorithm

- Step 1. We start from vertex a
- Step 2. Highlight edge ab
- Step 2. Highlight edge bf
- Step 2. Cannot choose fa, fb, highlight edge fd
- Step 2. Highlight edge dg
- Step 2. Highlight edge gc
- Step 2. Highlight edge ce
- step 5. Close the cycle by adding ea

$$a \to b \to f \to d \to g \to c \to e \to a$$

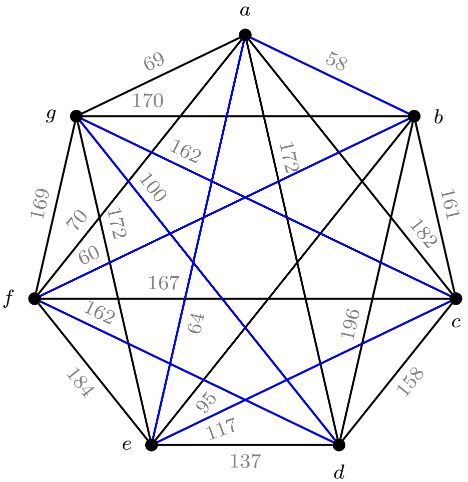


Nearest Neighbor Algorithm

- Step 1. We start from vertex a
- Step 2. Highlight edge ab
- Step 2. Highlight edge bf
- Step 2. Cannot choose fa, fb, highlight edge fd
- Step 2. Highlight edge dg
- Step 2. Highlight edge gc
- Step 2. Highlight edge ce
- step 5. Close the cycle by adding ea

$$a \to b \to f \to d \to g \to c \to e \to a$$

- Total weight: 723



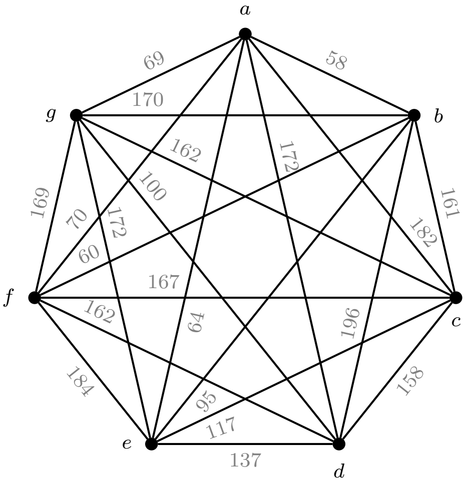
Cheapest Link Algorithm

1. Among all edges in the graph, pick the one with the smallest weight. If more than one possible choices have the same weight, randomly choose one. Highlight the chosen weight
2. Repeat step 1 with the added conditions
   - no vertex has three highlighted edges incident to it
   - no edge is chosen so that a cycle closes before hitting all the vertices
3. Calculate the total weight

Cheapest Link Algorithm

- Step 1. Choose edge ab
- Step 1. Choose edge bf
- Step 1. Choose edge ae
- Step 1. Cannot choose edge ag, otherwise a would have three edges incident to it. Choose edge dg
- Step 1. Choose edge ce
- Step 1. Choose edge cd
- Close the cycle fg

$$a \to b \to f \to g \to d \to c \to e \to a$$

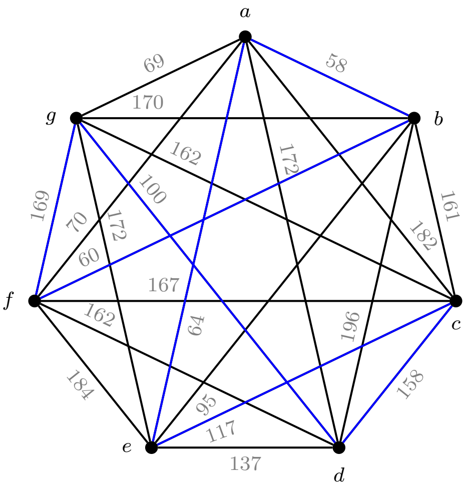


Cheapest Link Algorithm

- Step 1. Choose edge ab
- Step 1. Choose edge bf
- Step 1. Choose edge ae
- Step 1. Cannot choose edge ag, otherwise a would have three edges incident to it. Choose edge dg
- Step 1. Choose edge ce
- Step 1. Choose edge cd
- Close the cycle fg

$$a \to b \to f \to g \to d \to c \to e \to a$$

- Total weight: 726



Nearest Insertion Algorithm

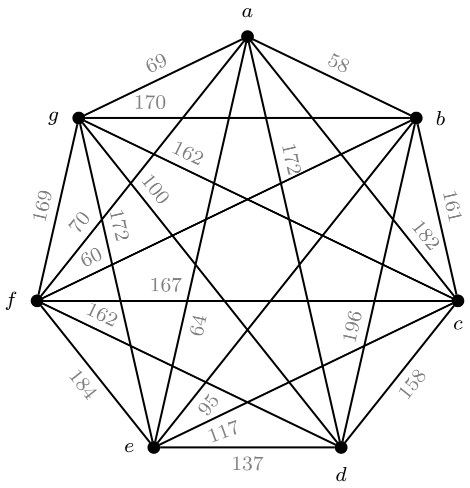
1. Among all edges in the graph, pick one with the smallest weight. Highlight the edge and its endpoints.
2. Pick a vertex that is closest to one of the two already chosen vertices. Highlight the new vertex and its edges to both of the previously chosen vertices.
3. Pick the vertex that is closest to any of the already chosen vertices. Insert this vertex into the existing cycle by connecting it to the nearest chosen vertex. Then, add a second edge to complete the insertion and remove one existing edge to maintain a cycle. Choose the scenario with the smallest total weight.
4. Repeat step 3 until all vertices have been included in the cycle
5. Calculate the total weight

Nearest Insertion Algorithm

- Step 1. Choose edge ab
- Step 2. Pick vertex f, add edges fb, fa.
- Step 3. Pick vertex e, add edge be. We need to decide if add edge ae and delete ab or add fe and delete bf

$$\begin{aligned} \omega(ae) - \omega(ab) &= 64 - 58 = 6 \\ \omega(fe) - \omega(bf) &= 184 - 60 = 124 \end{aligned}$$

Hence we add ae and delete ab

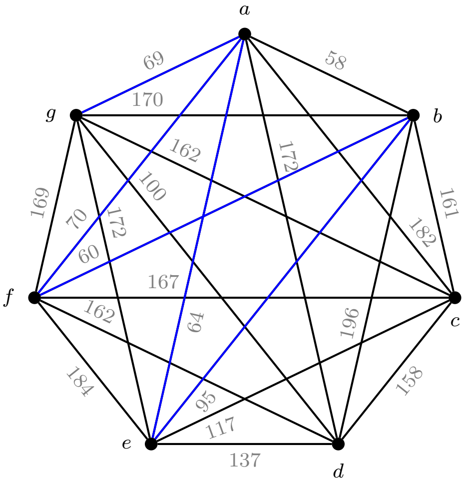


Nearest Insertion Algorithm

- Step 3. Pick vertex g, add edge ga. We need to decide if add edge gf and delete af or add ge and delete ae

$$
\begin{aligned}
\omega(gf) - \omega(af) &= 169 - 70 = 99 \\
\omega(ge) - \omega(ae) &= 172 - 64 = 108
\end{aligned}
$$

Hence we add gf and delete af

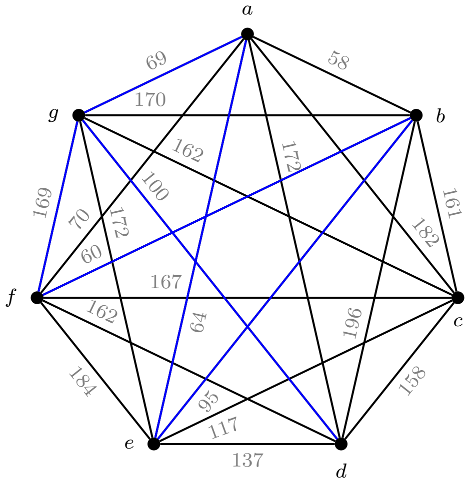


Nearest Insertion Algorithm

- Step 3. Pick vertex d, add edge gd.
  We need to decide if add edge da and delete ag or add df and delete gf

  $$\begin{aligned}\omega(da) - \omega(ga) &= 172 - 69 = 103 \\ \omega(df) - \omega(gf) &= 162 - 169 = -7\end{aligned}$$

  Hence we add df and delete gf

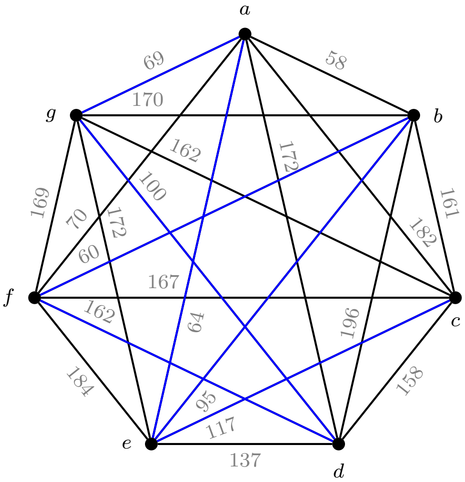


Nearest Insertion Algorithm

- Step 3. Pick vertex c, add edge ce. We need to decide if add edge ca and delete ea or add cb and delete eb

$$\begin{aligned} \omega(ca) - \omega(ea) &= 182 - 64 = 118 \\ \omega(cb) - \omega(eb) &= 161 - 95 = 66 \end{aligned}$$

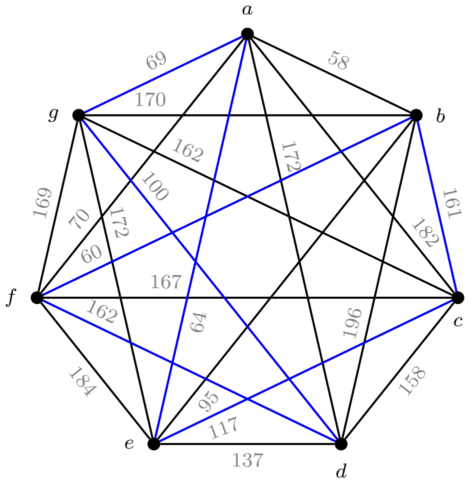
Hence we add cb and delete eb



Nearest Insertion Algorithm

$$a \to g \to d \to f \to b \to c \to e \to a$$

Total weight: 733



Undirecting Algorithm

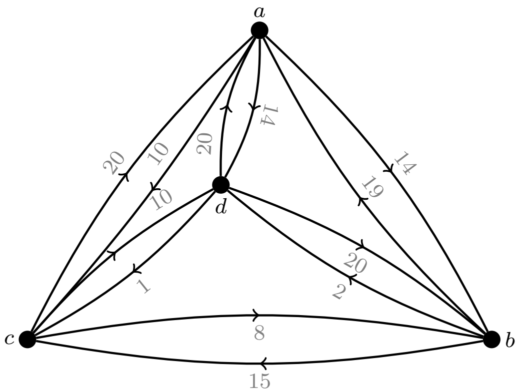
Input: Weighted complete digraph $G = (V, A, \omega)$
Steps:

1. For each vertex x, make a clone x'. Form the edge xx' with weight 0
2. For each arc xy form the edge $x'y$
3. The weight of an edge is equal to the weight of its corresponding arc
   - $\omega(xx') = 0$
   - $\omega(x'y) = \omega(xy)$
   - $\omega(xy') = \omega(yx)$

Output: weighted clone graph $G' = (V', E, \omega')$

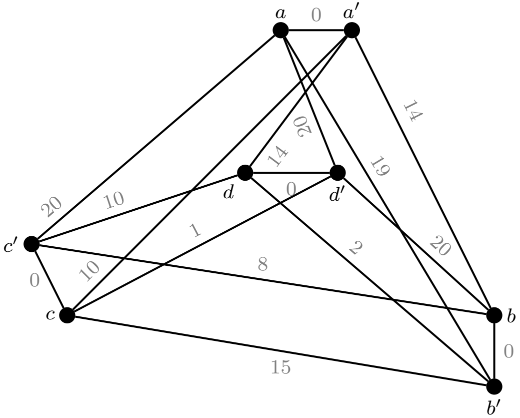
Undirecting Algorithm

	a	b	c	d
a	.	14	10	14
b	19	.	15	2
c	20	8	.	10
d	20	20	1	.

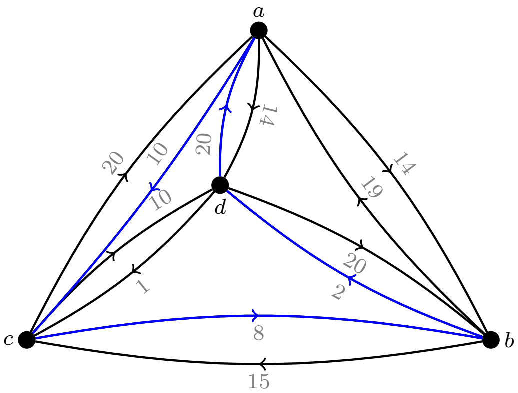


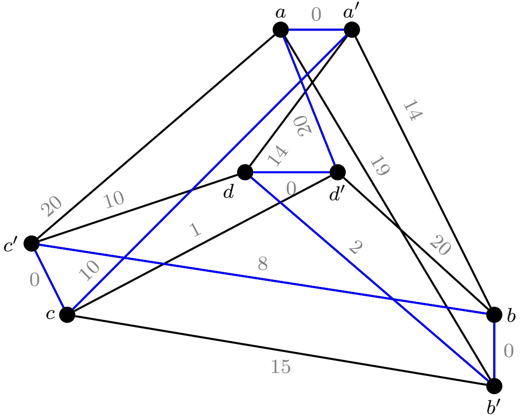
Undirecting Algorithm

	a	b	c	d	a'	b'	c'	d'
a	.	.	.	.	0	19	20	20
b	.	.	.	.	14	0	8	20
c	.	.	.	.	10	15	0	1
d	.	.	.	.	14	2	10	0
a'	0	14	10	14	.	.	.	.
b'	19	0	15	2	.	.	.	.
c'	20	8	0	10	.	.	.	.
d'	20	20	1	0	.	.	.	.



Nearest Neighbor Algorithm





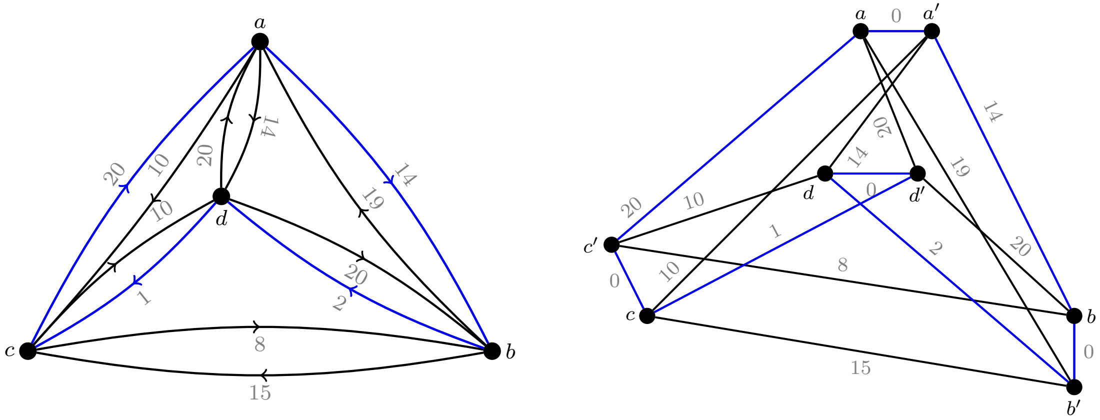
- Nearest Neighbor Algorithm

$$aa'cc'bb'dd'a$$

- Convert to the digraph

$$a \to c \to b \to d \to a$$

Cheapest Link Algorithm



- Cheapest Link Algorithm

$$aa'bb'dd'cc'a$$

- Convert to the digraph

$$a \to b \to d \to c \to a$$