Tutorial 9

Hamiltonian cycles

Question 1. Compute the following factorials

1. 8!
2. 12!
3. 16!

Solution.

1. $40,320$
2. $479,001,600$
3. $20,922,789,888,000$

Question 2. Simplify the following factorials:

1. $9 \times 8!$
2. $\frac{11!}{8!}$
3. $6! \times \frac{7!}{5!}$

Solution.

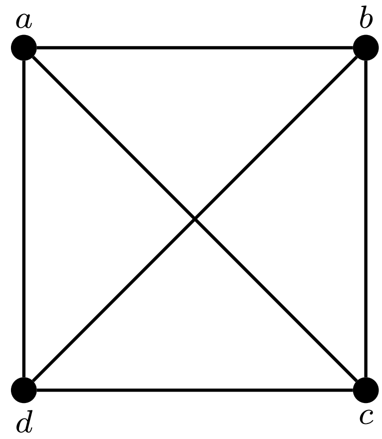
1. $9! = 362,880$
2. $11 \times 10 \times 9 = 990$
3. $7! \times 6 = 30,240$

Question 3. How many different Hamiltonian cycles are there for K_4? K_8? K_{10}? Draw all possible Hamiltonian cycles for K_4.

Solution.

- K_4: $(4-1)! = 3! = 6$
- K_8: $(8-1)! = 7! = 5040$
- K_{10}: $(10-1)! = 9! = 362,880$

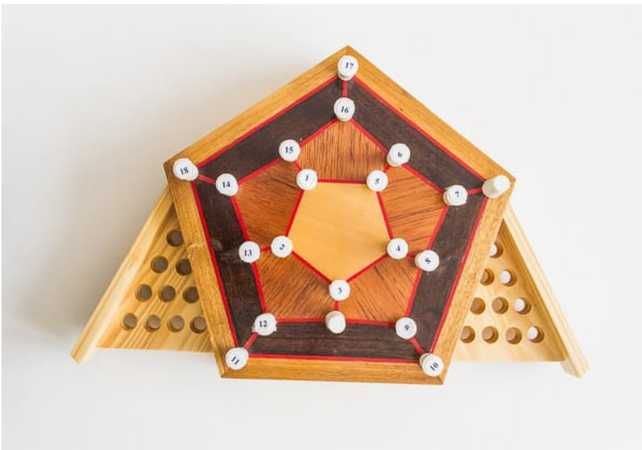
The graph of K_4 is as follows



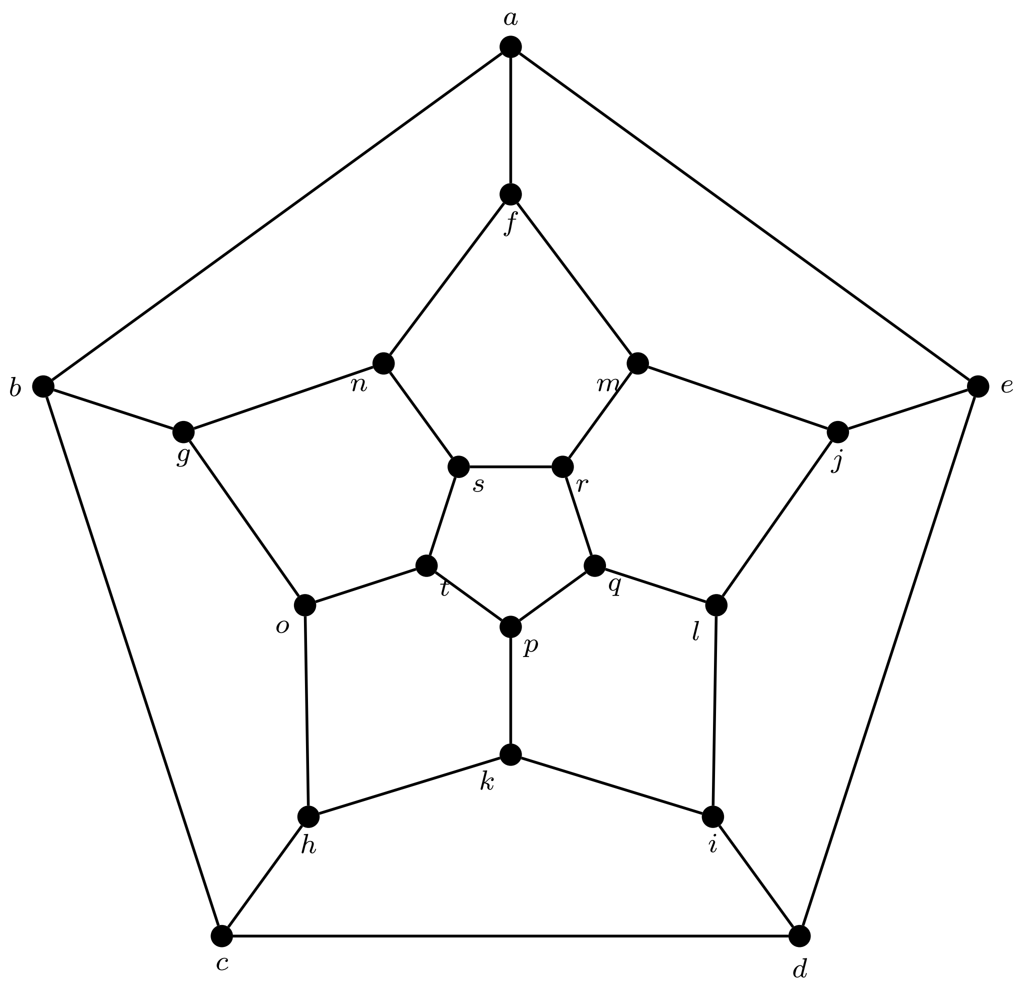
Taking a to be the reference point, the possible Hamiltonian cycles are

1. $abcda$, $adcba$
2. $abdca$, $acdba$
3. $acbda$, $adbca$

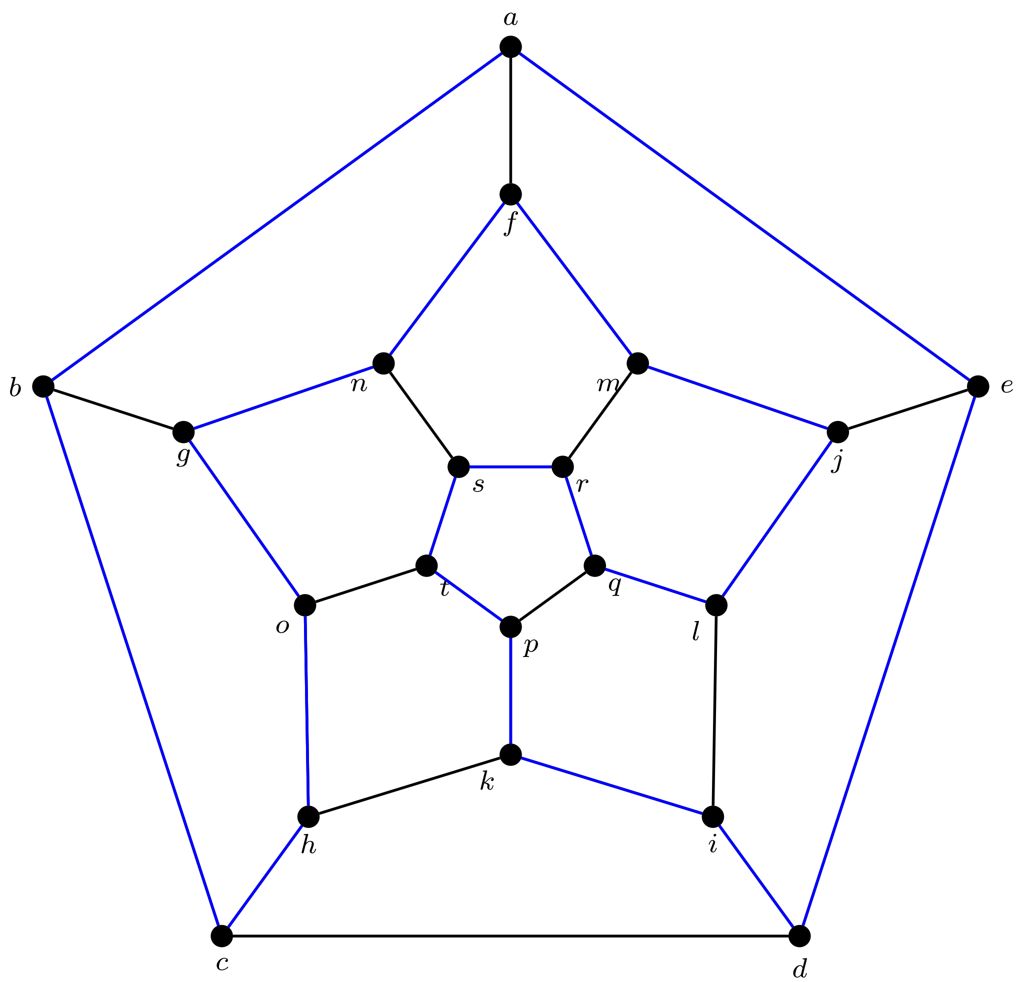
Question 4. Sir William Hamilton formalized the ideas of the Hamiltonian cycle and path. He posed this idea in 1856 in terms of a puzzle, which he later sold to a game dealer. The "Icosian Game" was a wooden puzzle with numbered ivory pegs where the player was tasked with inserting the pegs so that following them in order would traverse the entire board (as shown in the figure below).



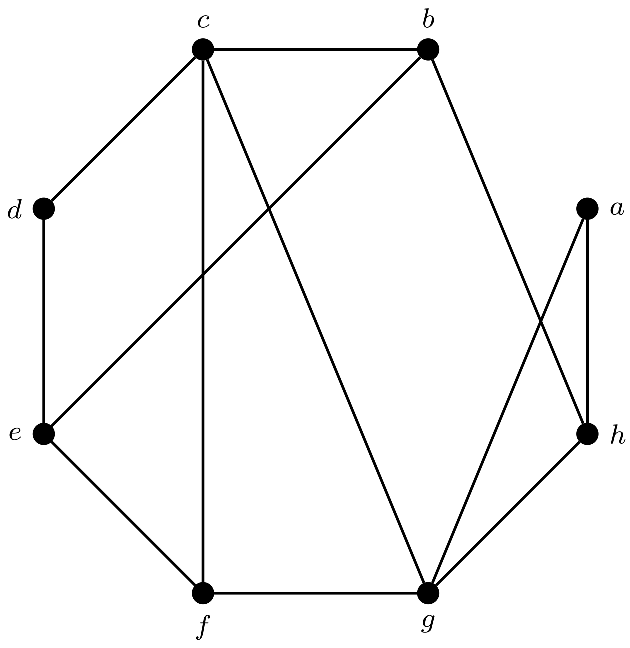
This is equivalent to finding a Hamiltonian cycle in the following graph. Solve the problem.



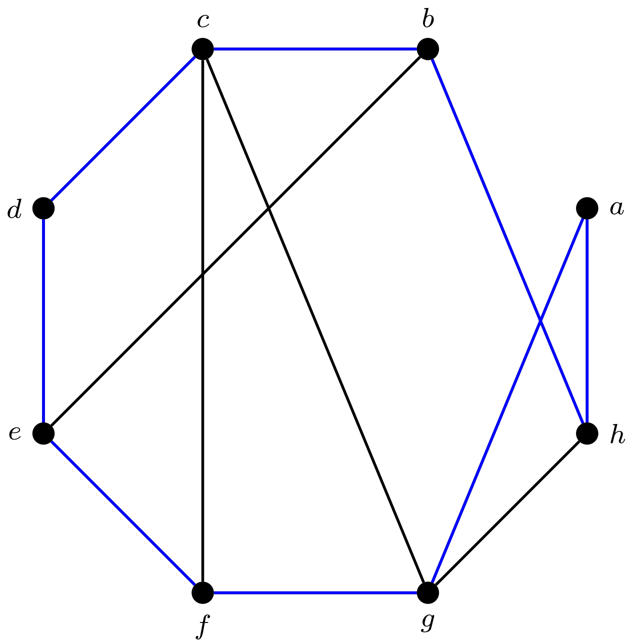
Solution.



Question 5. Find a Hamiltonian cycle for the following graph.



Solution.



Question 6. For each of the graphs below, determine if G

(a) definitely has a Hamiltonian cycle;

(b) definitely does not have a Hamiltonian cycle; or

(c) may or may not have a Hamiltonian cycle.

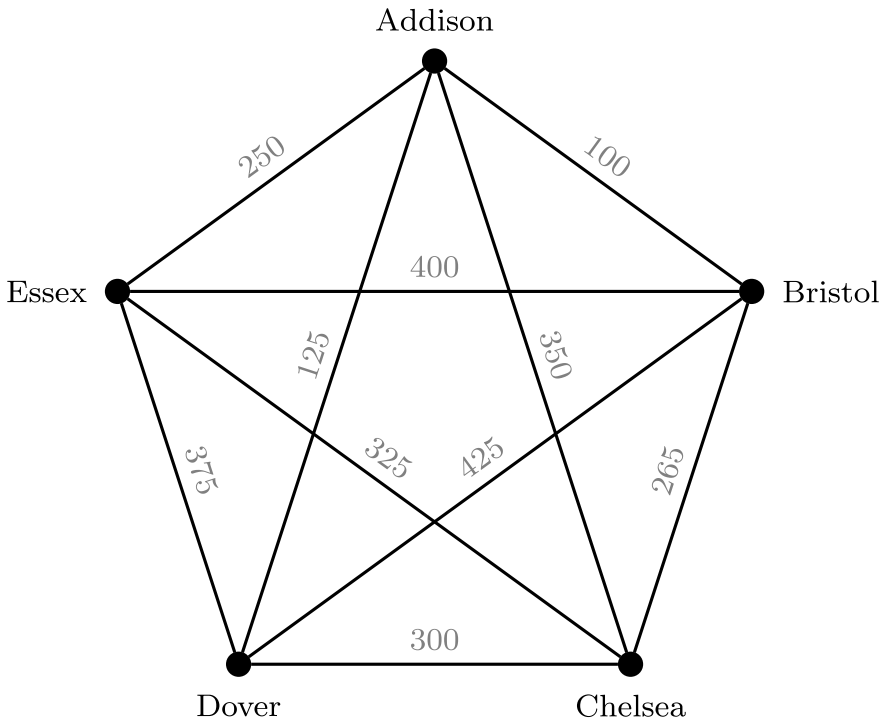
Explain your answer.

1. G has vertices of degree $3, 3, 3, 4, 4, 5$.
2. G is connected with 10 vertices, all of which have degree 6.
3. G has vertices of degree $1, 2, 2, 3, 5, 5$.
4. G is connected with vertices of degree $2, 2, 3, 3, 4, 4$.
5. G has vertices of degree $0, 2, 2, 4, 4, 5, 5$.

Solution.

1. (c). G has 6 vertices, all of which have degree at least 3. If G is connected, then it satisfies Dirac's Theorem and has a Hamiltonian cycle. If G is not connected, then it does not have a Hamiltonian cycle
2. (a). By Dirac's Theorem
3. (b). G cannot have a Hamiltonian cycle if it contains a vertex of degree one
4. (c). It is possible though not guaranteed
5. (b). G is not connected

Question 7. Apply Repetitive Nearest Neighbor to the following graph.



Solution.

- $abcdea - 1290$
- $badceb \to adceba - 1250$
- $cbadec \to adecba - 1190$
- $dabced \to abceda - 1190$
- $eabcde \to abcdea - 1290$

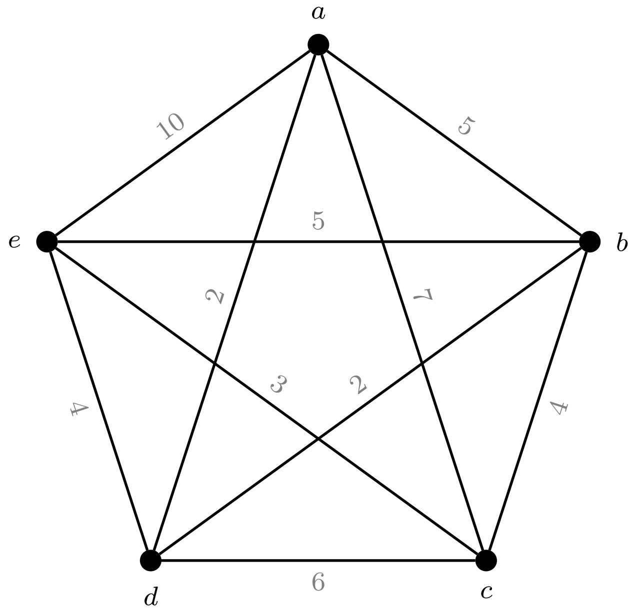
Question 8. Find a Hamiltonian cycle for each of the graphs below using:

(i) Repetitive Nearest Neighbor,

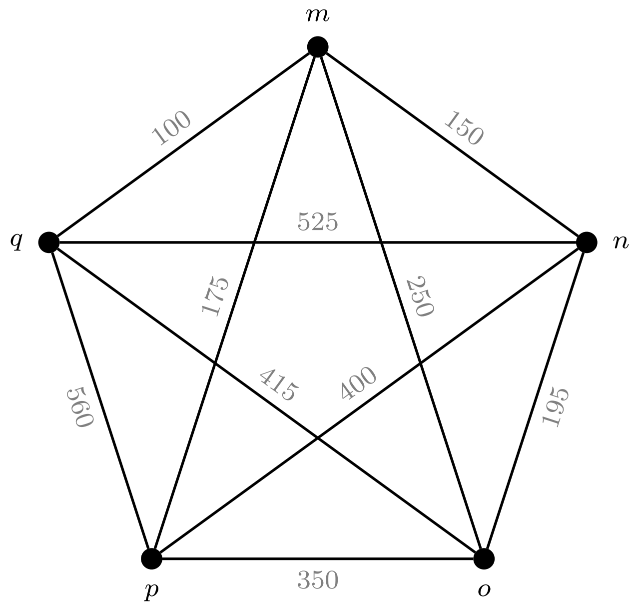
(ii) Cheapest Link, and

(iii) Nearest Insertion.

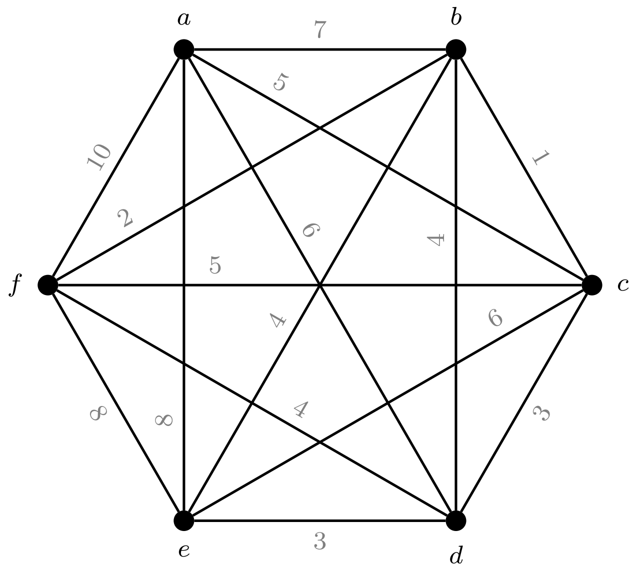
1.


a
b
c
d
e
10
5
5
2
7
4
4
3
2
6

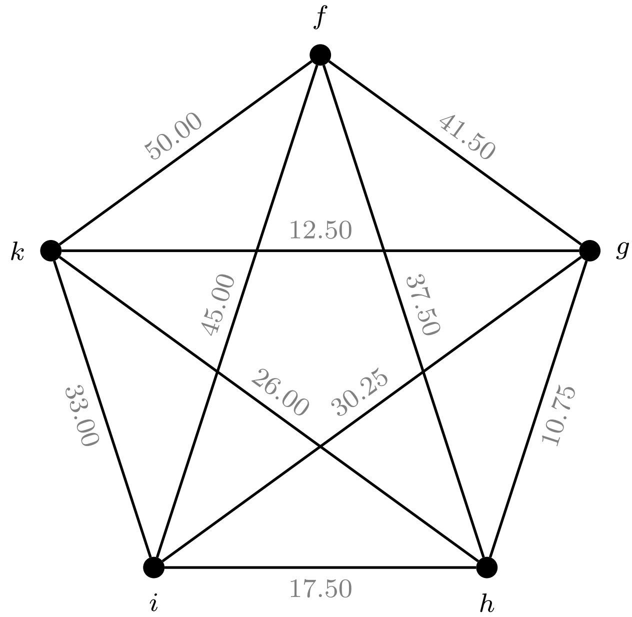
2.


m
n
o
p
q
100
150
525
175
250
560
195
415
400
350

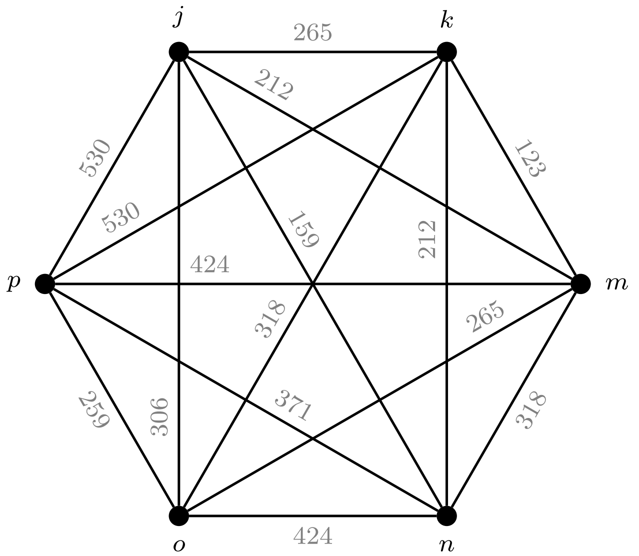
3.


a
b
c
d
e
f
7
5
10
1
2
6
4
5
4
6
8
8
4
3
3

4.


f
g
h
i
k
50.00
41.50
12.50
45.00
37.50
33.00
26.00
30.25
10.75
17.50

5.


j
k
m
n
o
p
265
212
530
123
530
159
212
424
318
265
259
306
371
318
424

Solution.

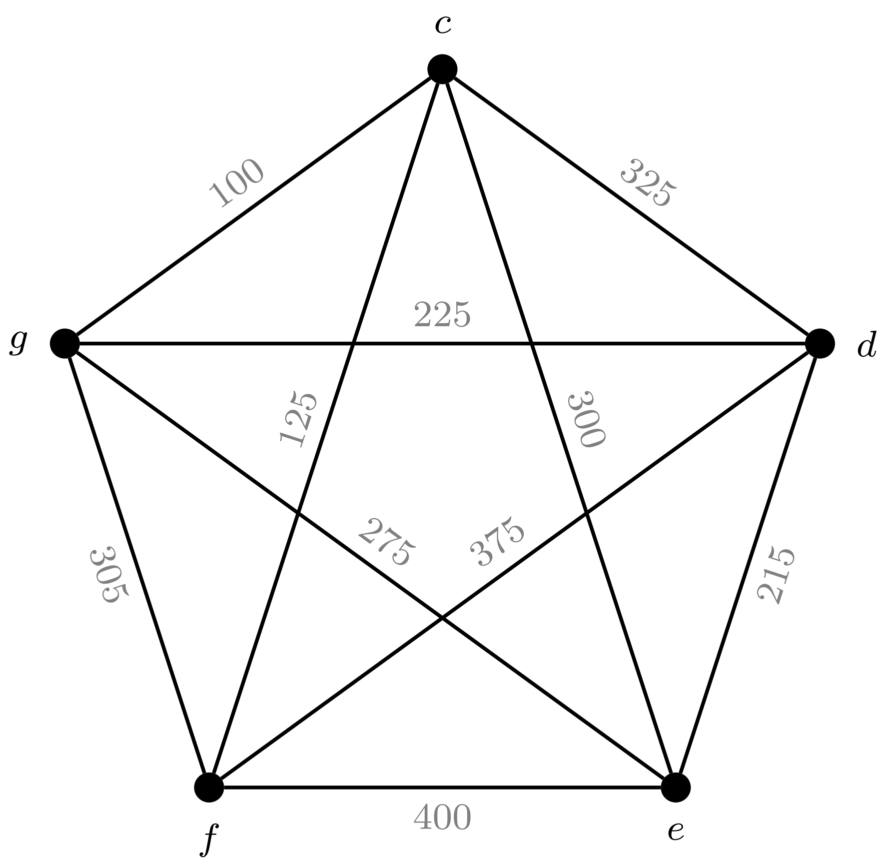
1. (i) $adbcea$, 21; $bdaceb \to acebda$, 19; $cedabc \to abceda$, 18; $dabced \to abceda$, 18; $ecbade \to abceda$, 18
   (ii) $adbcea$, 21
   (iii) $abceda$, 18
2. (i) $mqonpm - 1285$; $nmqopn - 1415$; $onmqpo - 1355$; $pmqonp - 1285$; $qmnopq - 1355$
   (ii) $mnopqm - 1355$
   (iii) $mpnoqm - 1285$
3. (i) $acbfdea - 23$; $bcdeafb - 27$; $cbfdeac - 23$; $debcafd - 27$; $edcbfae - 27$; $fbcdeaf - 27$ (answers may vary in the case of ties)
   (ii) $aedcbfa - 27$
   (iii) $acbfdea - 23$
4. (i) $fhgkif - 138.75$; $ghikfg - 152.75$; $hgkifh - 138.75$; $ihgkfi - 135.75$; $kghifk - 135.75$
   (ii) $fihgkf - 135.75$
   (iii) $fhgkif - 138.75$
5. (i) $jnkmopj - 1548$; $kmjnpok - 1442$; $mknjopm - 1483$; $njmkopn - 1442$; $opnjmko - 1351$; $pomknjp - 1548$
   (ii) $jmkopnj - 1442$
   (iii) $jnkmpoj - 1483$

Question 9. Chris wants to visit his 4 brothers over the holidays and has determined the costs as shown in the table below. Find a route (and its total weight) for Chris using

1. Repetitive Nearest Neighbor
2. Cheapest Link
3. Nearest Insertion

	Chris	David	Evan	Frank	George
Chris	.	325	300	125	100
David	325	.	215	375	225
Evan	300	215	.	400	275
Frank	125	375	400	.	305
George	100	225	275	305	.

Solution.



1. Repetitive Nearest Neighbor
   - $cgdefc$ – 1065
   - $degcfd \to cfdegc$ – 1090
   - $fcgdef \to cgdefc$ – 1065
   - $gcfdeg \to cfdegc$ – 1090

   Thus, the best route is either $cgdefc$ or $cgdefc$ both with weight 1065.

2. Cheapest Link: $cgdefc$– 1065

3. Nearest Insertion: $cfedgc$ – 1065

Question 10. June and Tori are planning their annual winery tour of Virginia. They want to plan their route so they can see as many of the wineries in one day as possible and this year will be staying at the inn at Mt. Eagle Winery. The chart below lists the wineries and the time (in minutes) between each one. Find a possible route (and its total time) for June and Tori using

1. Repetitive Nearest Neighbor

2. Cheapest Link

3. Nearest Insertion

4. and determine if they can visit all six locations in one day.

	Bluebird Wines	Cardinal Winery	Elk Point Vineyard	Red Fox Wines	Graybird Vineyard	Mt. Eagle Winery
Bluebird	.	41	58	43	51	49
Cardinal	41	.	60	7	62	33
Elk Point	58	60	.	75	67	53
Red Fox	43	7	75	.	64	36
Graybird	51	62	67	64	.	68
Mt. Eagle	49	33	53	36	68	.

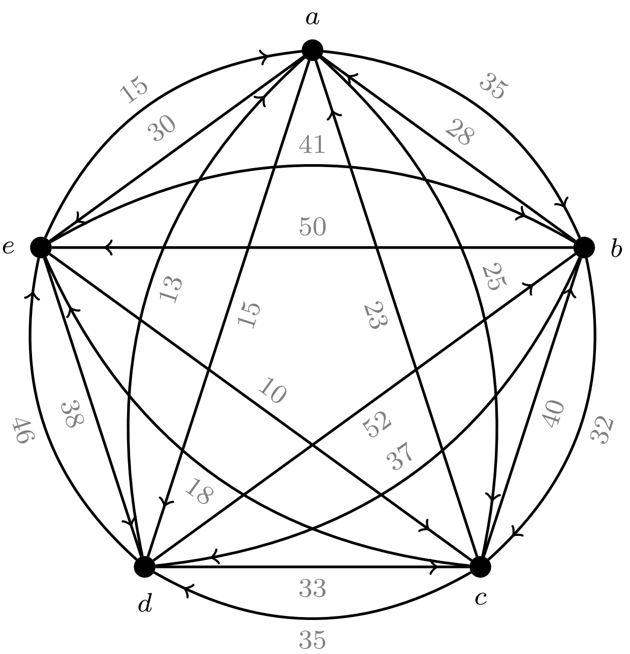
Solution.

1. Repetitive Nearest Neighbor
   - $bcrmegb \to megbcrm$ – 255
   - $crmbgec \to mbgecrm$ – 270
   - $emcrbge \to mcrbgem$ – 254
   - $rcmbger \to mbgercm$ – 282
   - $gbcrmeg \to megbcrm$ – 255
   - $mcrbgem$– 254

   the best route is $mcrbgem$, which takes 254 minutes
2. Cheapest Link: $crbgemc \to mcrbgem$ – 254
3. Nearest Insertion: $brcmegb \to megbrcm$ – 254
4. The shortest route takes 254 minutes, which is 4hours and 14 minutes. Hence they can visit all six wineries in one day.

Question 11. Using the digraph below,

1. Apply the Undirecting Algorithm to find the weighted clone graph.
2. Using your result from 1, apply the Nearest Neighbor Algorithm with starting vertices a, a', c and c' and convert your results to directed cycles in the digraph. Find the total weight of each directed cycle.
3. Using your result from 1, apply the Cheapest Link Algorithm and convert your result to a directed cycle in the digraph and find its total weight.



Solution.

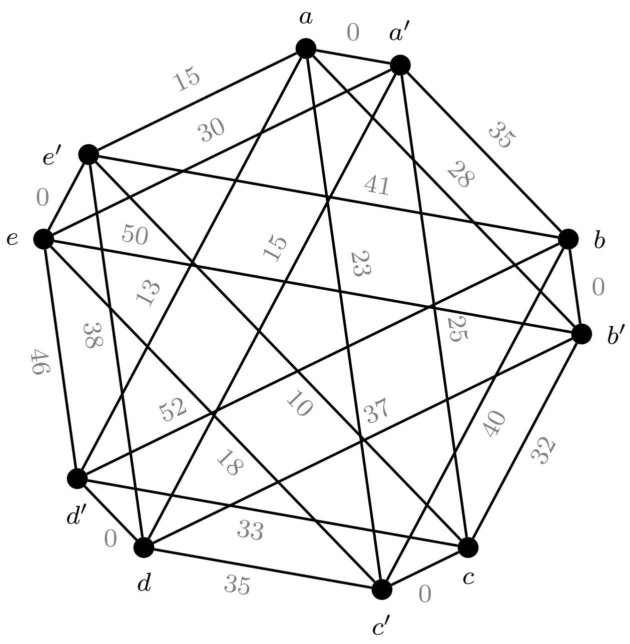
1. We first construct the table of edge weights

	a	b	c	d	e
a	.	35	25	15	30
b	28	.	32	37	50
c	23	40	.	35	18
d	13	52	33	.	46
e	15	41	10	38	.

Then we can construct the table of edge weights for the undirected clone graph as follows

	a	b	c	d	e	a'	b'	c'	d'	e'
a	.	.	.	.	.	0	28	23	13	15
b	.	.	.	.	.	35	0	40	52	41
c	.	.	.	.	.	25	32	0	33	10
d	.	.	.	.	.	15	37	35	0	38
e	.	.	.	.	.	30	50	18	46	0
a'	0	35	25	15	30	.	.	.	.	.
b'	28	0	32	37	50	.	.	.	.	.
c'	23	40	0	35	18	.	.	.	.	.
d'	13	52	33	0	46	.	.	.	.	.
e'	15	41	10	38	0	.	.	.	.	.

The weighted clone graph in graph representation is as follows



2. Nearest Neighbor Algorithm

Nearest Neighbor Cycle	Conversion	Total Weight
$aa'dd'cc'ee'bb'a$	$a \to d \to c \to e \to b \to a$	135
$a'ad'dc'ce'eb'ba'$	$a \to b \to e \to c \to d \to a$	143
$cc'ee'aa'dd'bb'c$	$c \to e \to a \to d \to b \to c$	132
$c'ce'ea'ad'db'bc'$	$c \to b \to d \to a \to e \to c$	130

3. Cheapest Link Algorithm:

$$aa'ee'cc'bb'dd'a, \quad a \to e \to c \to b \to d \to a$$

total weight: 130

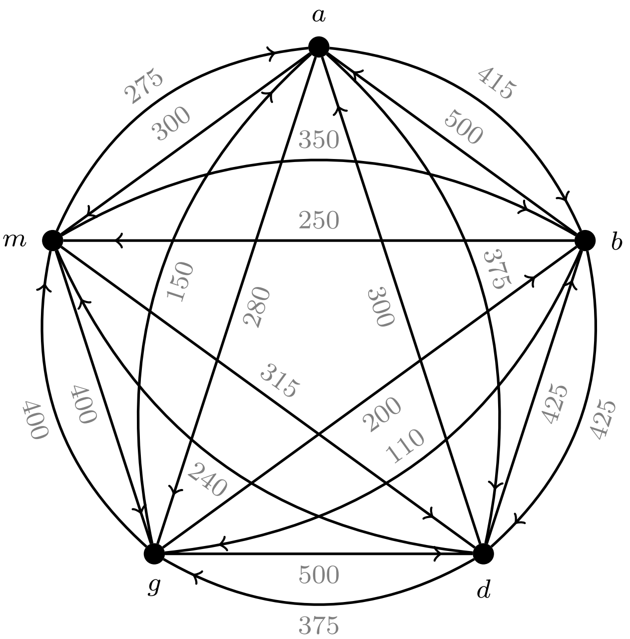
Question 12. Leena will be visiting her clients around Europe for the month of April. She has tried to estimate the cost of travel between two cities, using various modes of transportation and discovered the cost depends on the direction of travel. The table below gives these estimates.

1. Draw the directed graph representing the information in the chart below.
2. Apply the Undirecting Algorithm to find the weighted clone graph.
3. Using your result from 1, apply the Nearest Neighbor Algorithm with starting vertices a, a', d and d' and convert your results to directed cycles in the digraph. Find the total weight of each directed cycle.
4. Using your result from 1, apply the Cheapest Link Algorithm and convert your result to a directed cycle in the digraph and find its total weight.

	Amsterdam	Bern	Düsseldorf	Genoa	Munich
Amsterdam	.	415	375	280	300
Bern	500	.	425	110	250
Düsseldorf	300	425	.	375	240
Genoa	150	200	500	.	400
Munich	275	350	315	400	.

Solution.

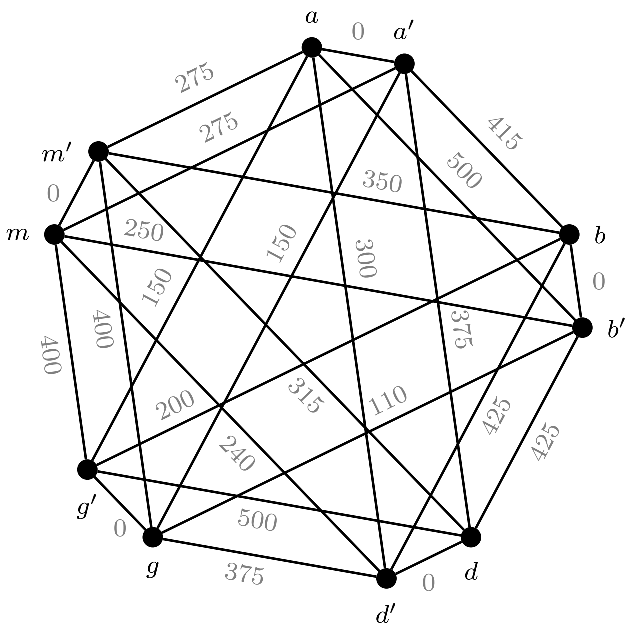
1. The directed graph representation is as follows



2. The table of weights for the weighted clone graph is as follows

	a	b	d	g	m	a'	b'	d'	g'	m'
a	.	.	.	.	.	0	500	300	150	275
b	.	.	.	.	.	415	0	425	200	350
d	.	.	.	.	.	375	425	0	500	315
g	.	.	.	.	.	280	110	375	0	400
m	.	.	.	.	.	300	250	240	400	0
a'	0	415	375	280	300	.	.	.	.	.
b'	500	0	425	110	250	.	.	.	.	.
d'	300	425	0	375	240	.	.	.	.	.
g'	150	200	500	0	400	.	.	.	.	.
m'	275	350	315	400	0	.	.	.	.	.

The graph representation is



3. Nearest Neighbor Algorithm

Nearest Neighbor Cycle	Conversion	Total Weight
$aa'gg'bb'mm'dd'a$	$a \to g \to b \to m \to d \to a$	1345
$a'ag'gb'bm'md'da'$	$a \to d \to m \to b \to g \to a$	1225
$dd'mm'aa'gg'bb'd$	$d \to m \to a \to g \to b \to d$	1420
$d'dm'mb'bg'ga'ad'$	$d \to a \to g \to b \to m \to d$	1345

4. Cheapest Link Algorithm:

$$aa'dd'mm'bb'gg'a, \quad a \to d \to m \to b \to g \to a$$

total weight: 1225

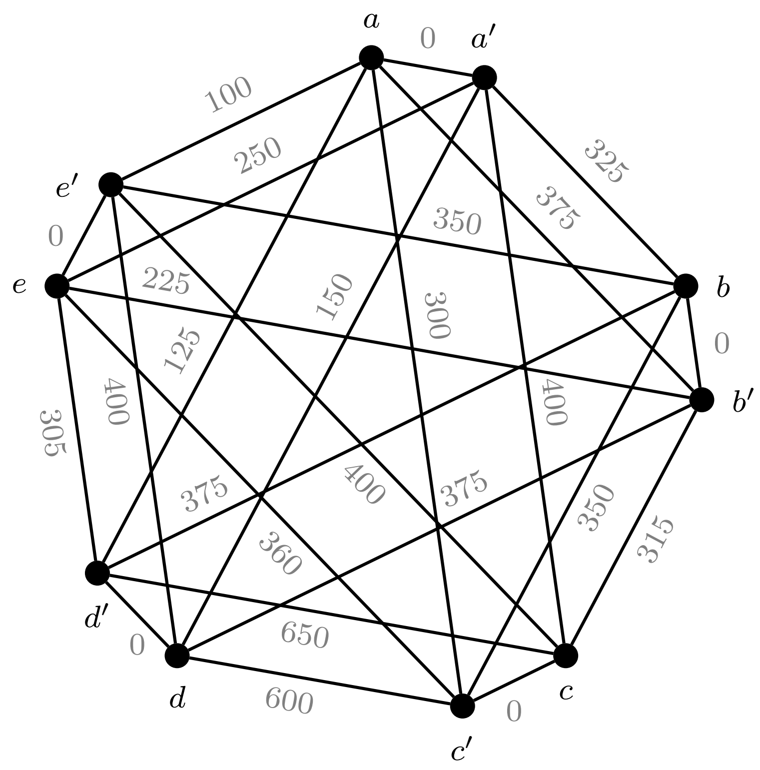
Question 13. Explain why no cycles of length three exist in the graph resulting from applying the Undirecting Algorithm to a complete digraph.

Solution. In the clone undirected graph constructed using the Undirecting Algorithm, each edge connects an original vertex to a clone vertex, and vice versa. Therefore, any path in the graph must alternate between original and clone vertices.

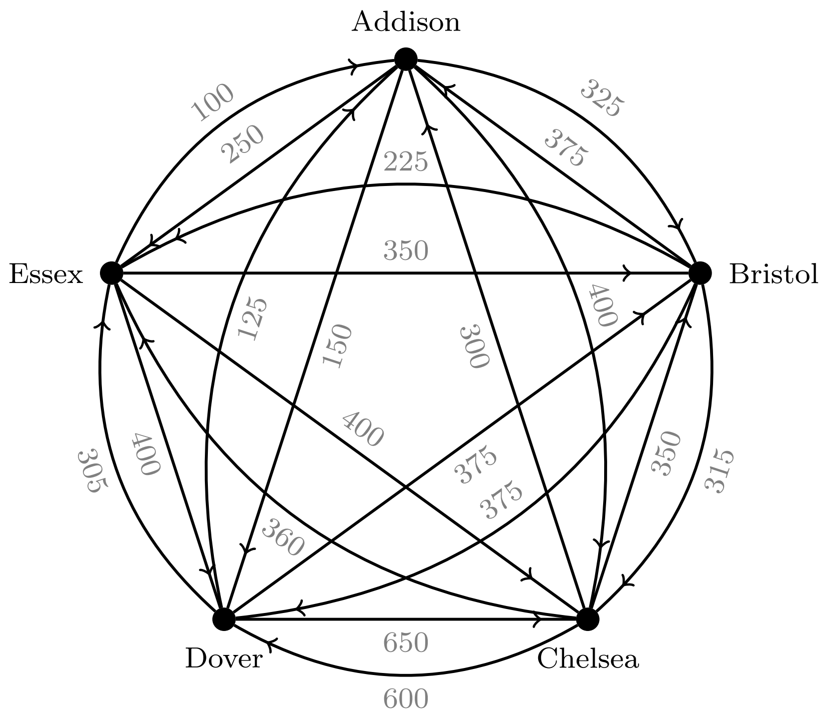
Suppose, for contradiction, that the graph contains a cycle of length three. Then the sequence of vertices in such a cycle must return to its starting point after two steps, implying that two of the vertices in the cycle must either both be original vertices or both be clone vertices.

Hence, the graph resulting from applying the Undirecting Algorithm to a complete directed graph contains no cycles of length three.

Question 14. Determine a modification of Nearest Insertion that will allow it to be used on a graph obtained from a complete digraph using the Undirecting Algorithm. (Hint: the initial cycle should start from the lowest nonzero edge and should have length 4.) Use your modification on the weighted clone graph



corresponding to the following directed graph from lecture



Solution. **Input**: Weighted clone graph of a complete digraph

1. **Initialization:**
    - Among all edges with nonzero weight, identify those with the minimum weight. If multiple such edges exist, select one at random.

- Use the selected edge to construct an initial cycle of length 4 by including both of its endpoints and their corresponding clone or original vertices.
- For example, if the chosen edge is $x'y$, the initial cycle is $x \to x'yy'x$. If the selected edge is xy', the initial cycle is $x'xy'yx'$.

2. **Insertion Step:**
   - Among all unvisited vertices, select the one that is closest to any vertex currently in the cycle.
   - Insert the selected vertex and its corresponding clone/original vertex into the cycle. This involves:
     - Connecting the chosen vertex to its nearest neighbor in the cycle,
     - Adding an edge between the chosen vertex and its clone/original counterpart,
     - Connecting the counterpart to a suitable position in the cycle, and
     - Removing one existing edge to maintain the cycle structure.
   - Choose the insertion configuration that results in the smallest increase in total cycle weight.
3. **Repeat** the insertion step until all original and clone vertices have been added to the cycle.
4. **Output:** The resulting Hamiltonian cycle in the clone graph corresponds to a Hamiltonian cycle in the original digraph. This can be obtained by interpreting the cycle as alternating between original and clone vertices.

Applying the algorithm to the weighted clone graph we get

$$ee'aa'dd'bb'cc'e$$

converting to directed cycle in the digraph we have

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

with total weight 1300.

Question 15. The Nearest Insertion Algorithm finds a Hamiltonian cycle by expanding smaller cycles through the addition of the closest vertex to that cycle. It suffers from the same problem as the other algorithms in that a large edge may be chosen in the last step of the algorithm. A variation, called *Farthest Insertion*, first considers the vertices farthest apart since any Hamiltonian cycle must include both of them. In doing so, later additions of vertices will either reduce the cycle weight or increase it by small margins. The description of the algorithm appears below.

Farthest Insertion Algorithm
Input: Weighted complete graph $G = (V, E)$.
Steps:

1. Pick a starting vertex v_1.
2. Choose the vertex v_2 that has the highest weighted edge to v_1.
3. Form a list $(w_1, w_2, w_3, \ldots, w_n)$ where the entry in location i is the minimum weighted edge from v_i to either of v_1 and v_2. The entries for v_1 and v_2 will be left blank (denoted by $-$).
4. Choose vertex x with the largest value from the list created in Step 3. Form the cycle $v_1\, v_2\, x\, v_1$.
5. Update the list from Step 3 so the entries are now the weights from the chosen to unchosen vertices. Choose the next vertex y with largest value in the list.
6. Append the cycle of chosen vertices with y by removing one of the edges from that cycle. Determine which edges to add and subtract by choosing the lowest total as in Nearest Insertion; that is, if the cycle obtained from Step 4 was $a - b - c - a$ and d is the new vertex to add along with edge to dc, we calculate:

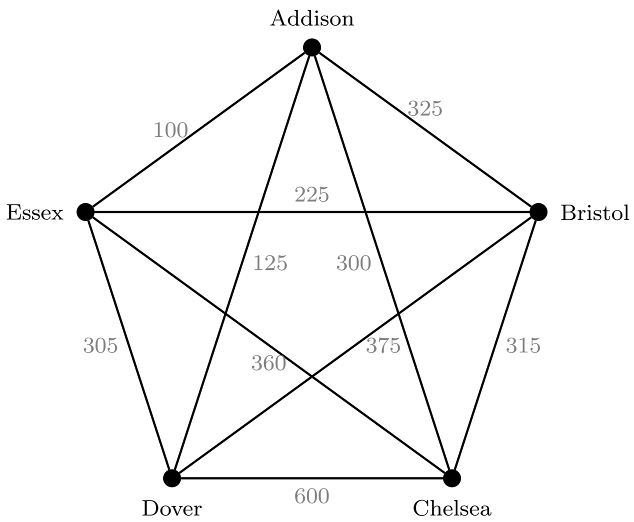
$$w(dc) + w(db) - w(cb) \quad \text{and} \quad w(dc) + w(da) - w(ca)$$

   and choose the option that produces the smaller total.
7. Repeat Steps (5) and (6) until all vertices have been included in the cycle.

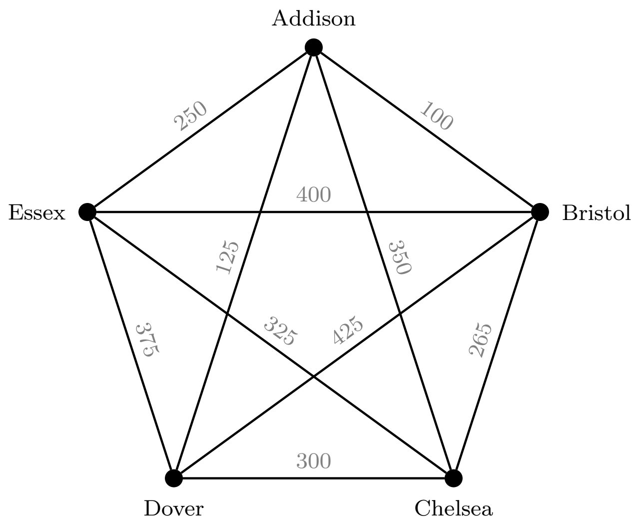
Output: Hamiltonian cycle.

Apply the Farthest Insertion to the following graph from the Lecture and from Question 7.

1.



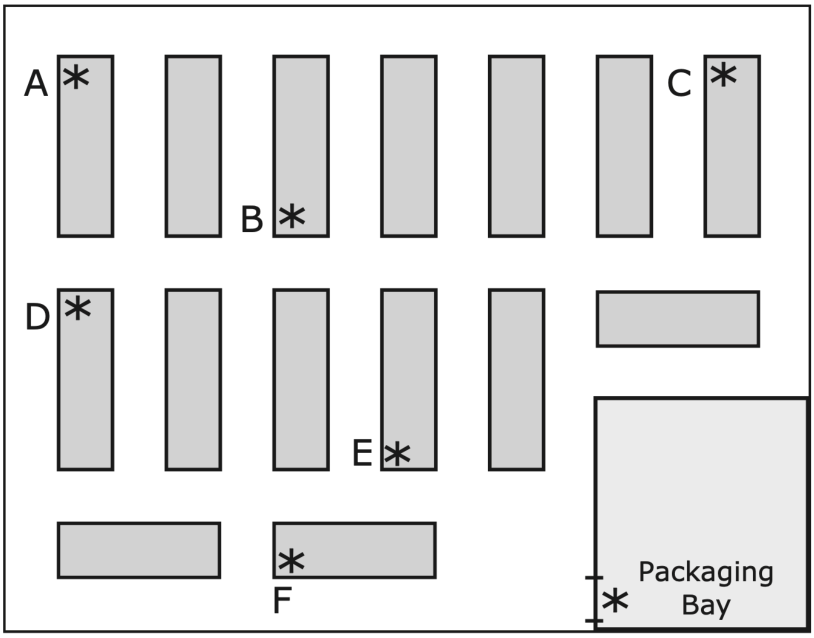
2.



Solution.

1. $adebca$ – 1270
2. $adceba$ – 1250

Question 16. Each morning a collection of online orders arrives at the warehouse of a large retailer. Steve, the warehouse manager, must ensure the items are packed and put onto the truck for shipment. However, the items are different every morning and are located in varying locations in the large warehouse. Steve has come to you for help in determining the best method for pulling stock from the shelves. Write a report detailing the Traveling Salesman Problem and how it applies to the warehouse. As part of your report, determine a route for the items shown in the map below. The route must start and end at the packaging bay (p) and the time required for moving down a long aisle is 45 seconds and down a short aisle or between aisles is 10 seconds. For example, it takes 85 seconds to get from item a to item b since four short segments and one long segment are used. Include a weighted graph and discussion of which algorithm(s) you used.



Solution. We can approximate the layout on a grid with the following coordinate for each point

$$a = \left(0,\ \ 4.5\right),\quad b = \left(4,\ \ 0\right),\quad c = \left(12,\ \ 4.5\right),\quad d = \left(0,\ \ -1\right),\quad e = \left(6,\ \ -5.5\right)$$

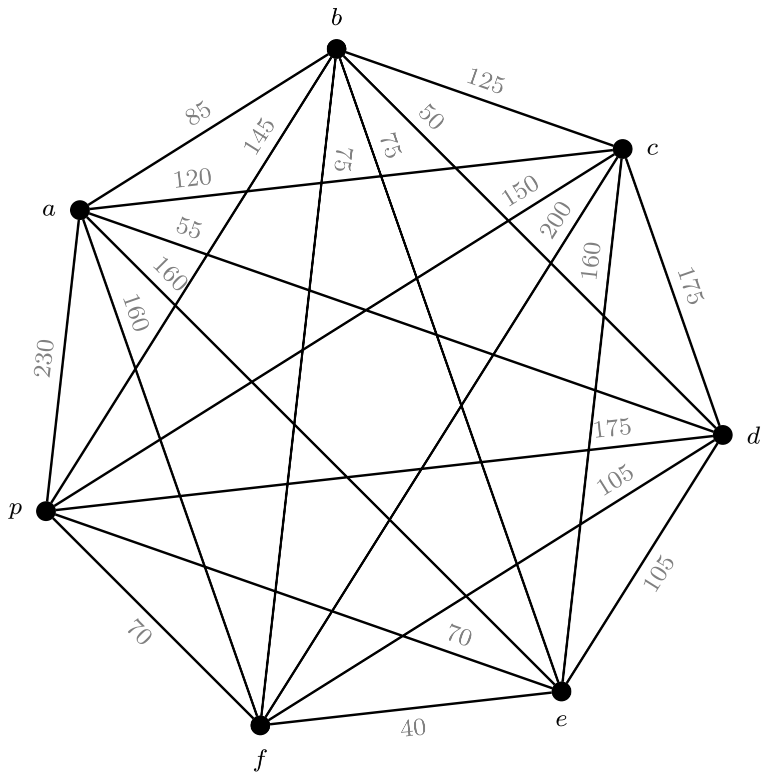
$$f = \left(4,\ \ -7.5\right),\quad p = \left(10,\ \ -8.5\right)$$

The weight for each edge is then given by

$\omega(ac) = 12 \times 10 = 120,$ $\omega(ad) = 5.5 \times 10 = 55,$ $\omega(ae) = (6 + 10) \times 10 = 160,$
$\omega(af) = (4 + 12) \times 10 = 160,$ $\omega(bc) = (8 + 4.5) \times 10 = 125,$ $\omega(bd) = (4 + 1) \times 10 = 50,$
$\omega(be) = (2 + 5.5) \times 10 = 75,$ $\omega(bf) = 7.5 \times 10 = 75,$ $\omega(cd) = (12 + 5.5) \times 10 = 175,$
$\omega(ce) = (6 + 10) \times 10 = 160,$ $\ldots$

	a	b	c	d	e	f	p
a	.	85	120	55	160	160	230
b	85	.	125	50	75	75	145
c	120	125	.	175	160	200	150
d	55	50	175	.	105	105	175
e	160	75	160	105	.	40	70
f	160	75	200	105	40	.	70
p	230	145	150	175	70	70	.

A graph representation is as follows



1. Repetitive Nearest Neighbor Algorithm
    - $adbefpca \to pcadbefp$ – 560
    - $bdacpefb \to pefbbdacp$ – 560
    - $cadbefpc \to pcadbefp$ – 560
    - $dbefpcad \to pcadbefp$ – 560
    - $efpbdace \to pbdacefp$ – 640
    - $fepbdacf \to pbdacfep$ – 680
    - $pefbdacp$ – 560
2. Cheapest Link Algorithm: $efbdacpe \to pefbdacp$ – 560
3. Nearest Insertion Algorithm: $pedacbfp$ – 620