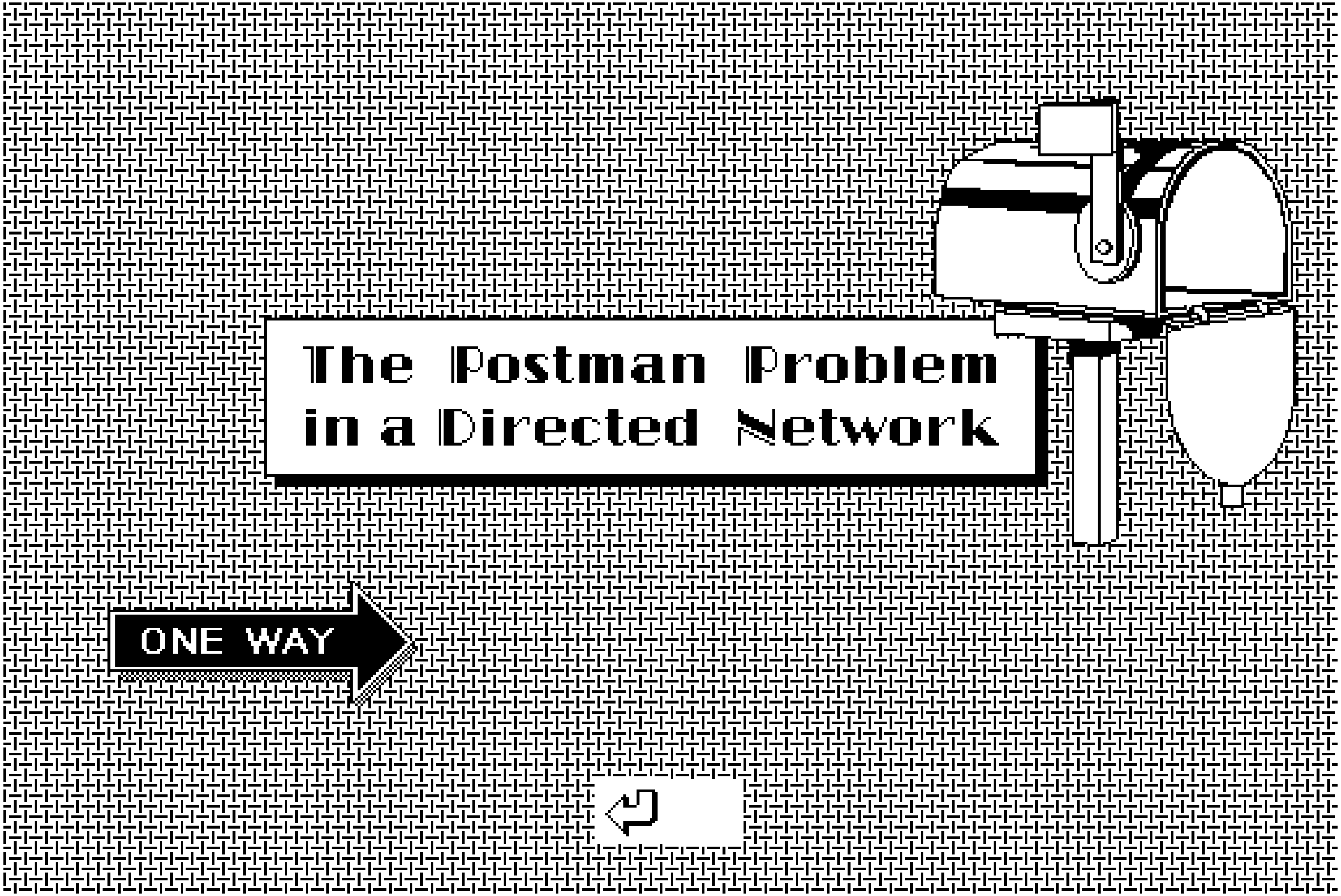
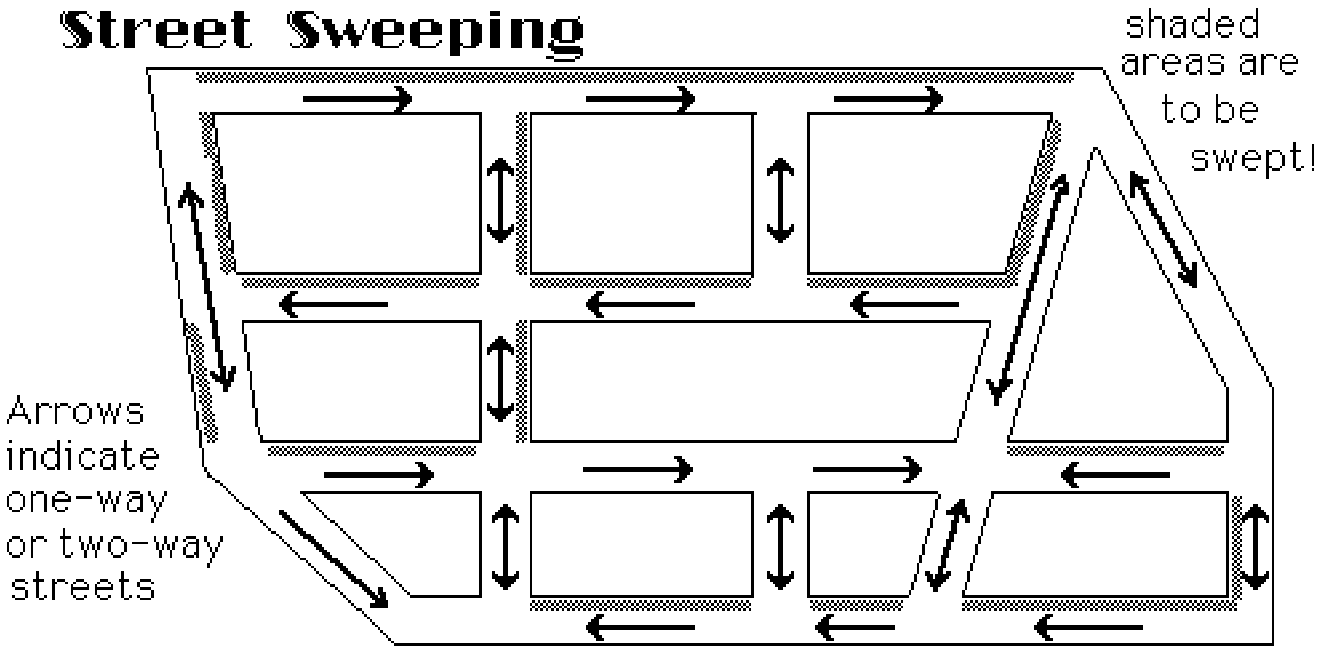

The Postman Problem
in a Directed Network
ONE WAY

Example: Street Sweeping

Plan a route for a street-sweeping machine which cleans all curbsides for which "No Parking" is currently in effect.

Some streets are one-way streets.
Street-sweeping machine must obey any one-way restrictions, and on two-way streets must travel on the right side, with the flow of traffic.

Street Sweeping



Travel times:

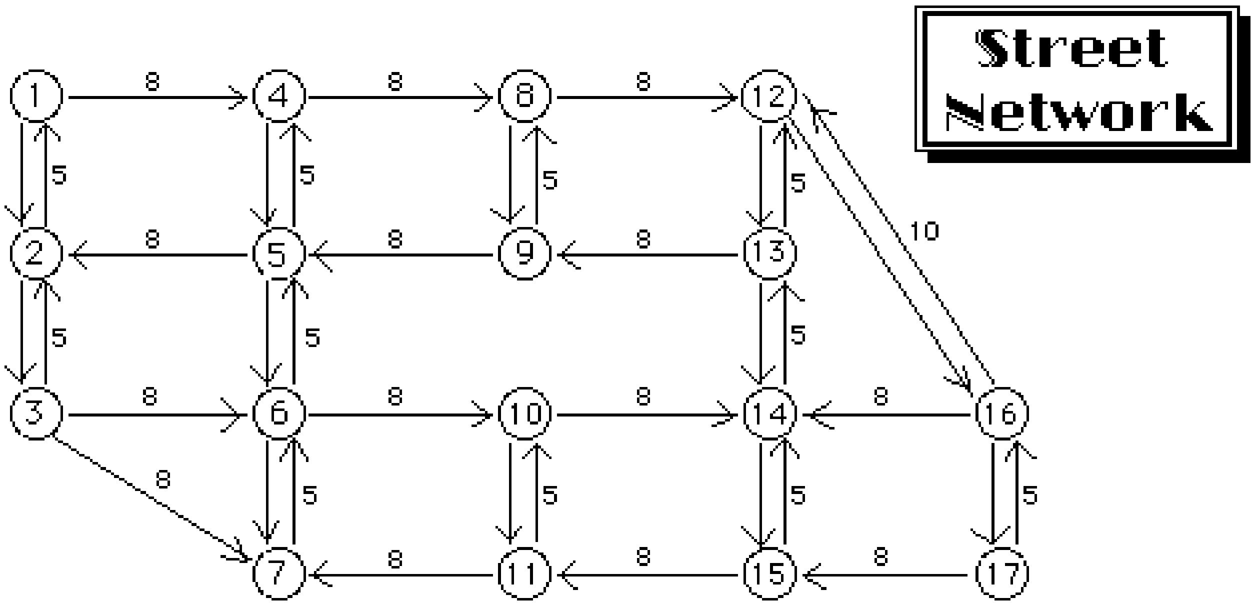
E-W: 8 minutes/block

N-S: 5 minutes/block

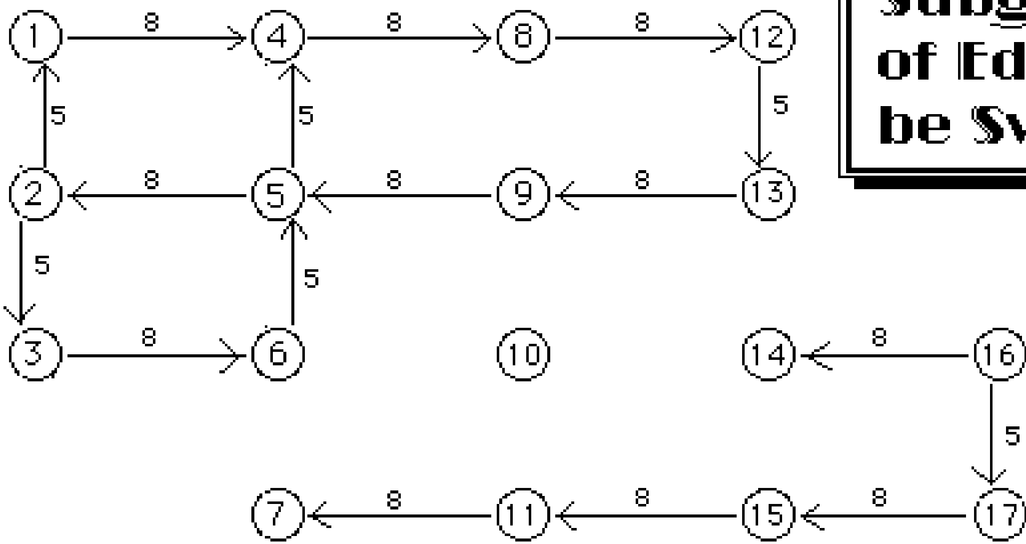
Diagonal blocks:

Lower left: 8 minutes

Upper right: 10 minutes


Street Network
1
4
8
12
2
5
9
13
3
6
10
14
16
7
11
15
17
8
5
10

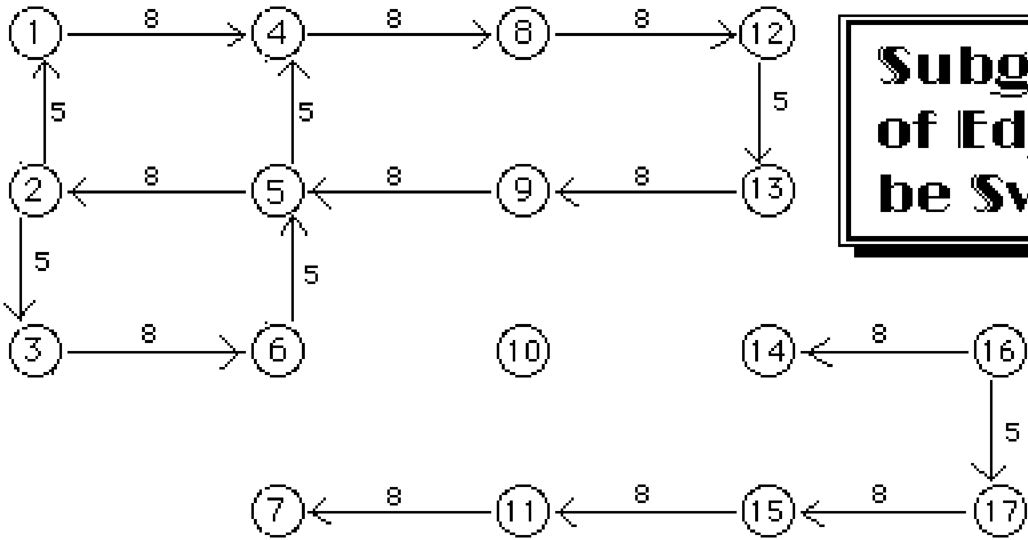
Subgraph of Edges to be Swept



Clearly no Euler tour exists for this subgraph (which is not connected)!

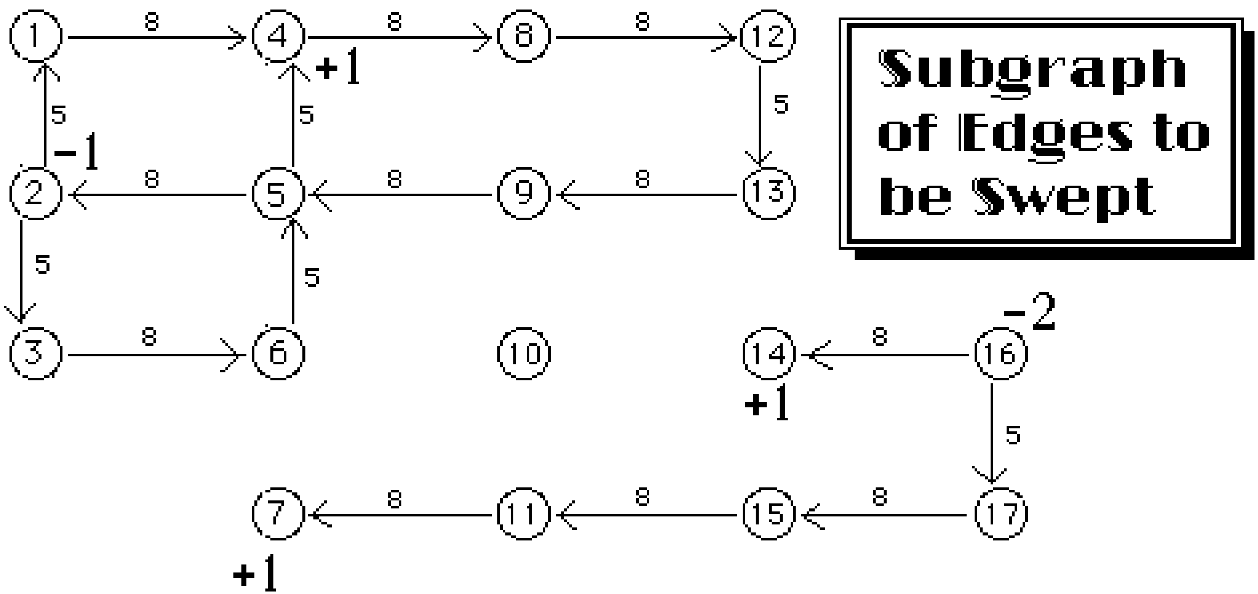
"Deadheading" will be necessary.

Subgraph of Edges to be Swept



Polarity = indegree − outdegree

Node:	2	4	7	14	16
Polarity:	−1	+1	+1	+1	−2



How should we add an appropriate set of "dead-heading" arcs so that an Euler tour can be constructed?

If a node has positive polarity, then more arcs enter the node than leave it.... so to make the polarity zero, we must add arcs leaving the node.

Conversely, if the node has negative polarity, then arcs must be added which enter the node in order to make the polarity zero.

Consider the positive polarities as "sources" of shipments to be made, and the negative polarities as "demands".

The "shipping cost" is the cost of adding a path from a "source" node to a "demand" node, i.e., the length of the shortest path between the nodes.

SOURCE \ DEMAND	#2	#16	supply
#4	13	26	1
#7	18	41	1
#14	29	20	1
rqmt	1	2	

Transportation Model

$$\text{Minimize } 13X_{4,2} + 26X_{4,16} + 18X_{7,2} + 41X_{7,16} + 29X_{14,2} + 20X_{14,16}$$

subject to

$$X_{4,2} + X_{4,16} = 1$$

$$X_{7,2} + X_{7,16} = 1$$

$$X_{14,2} + X_{14,16} = 1$$

$$X_{4,2} + X_{7,2} + X_{14,2} = 1$$

$$X_{4,16} + X_{7,16} + X_{14,16} = 2$$

$$X_{4,2} \geq 0,\ X_{7,2} \geq 0,\ X_{14,2} \geq 0,$$

$$X_{4,16} \geq 0,\ X_{7,16} \geq 0,\ X_{14,16} \geq 0$$

Optimal Solution:

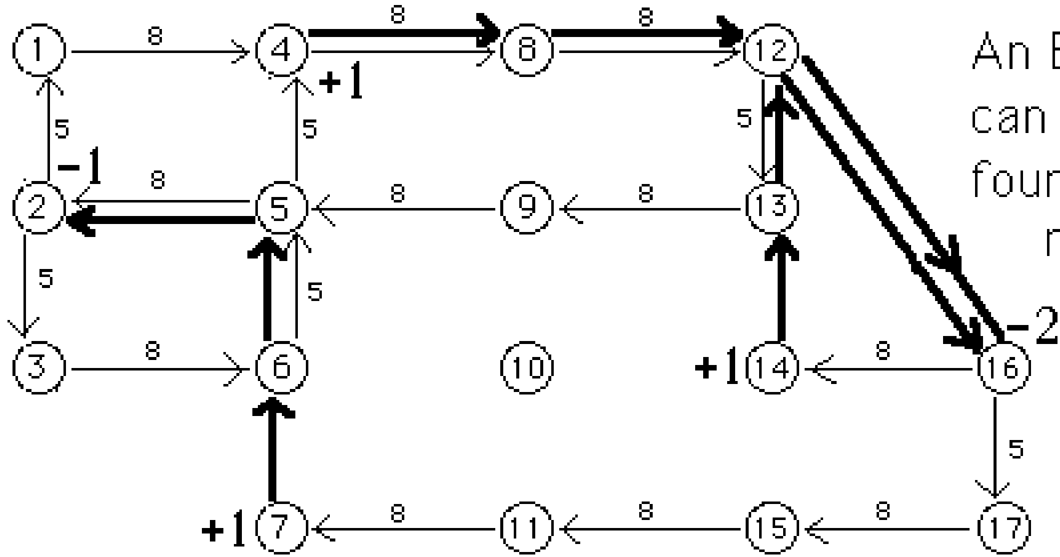
DEMAND

SOURCE	#2	#16	supply
#4	(13)	1 (26)	1
#7	1 (18)	(41)	1
#14	(29)	1 (20)	1
rqmt	1	2	

Transportation Model

That is,
add paths from:

node 4 to node 16
node 7 to node 2
node 14 to node 16



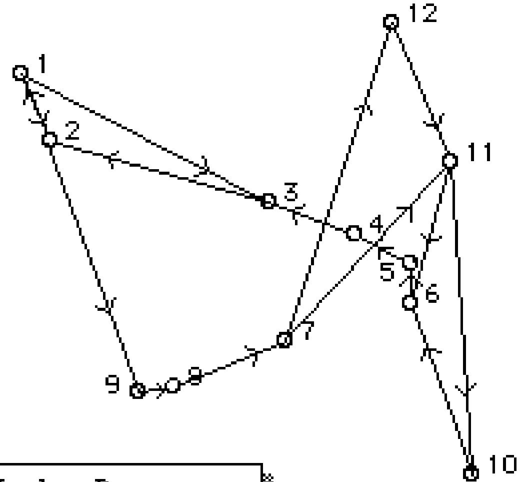
An Euler tour can now be found of the new network

The **bold** arcs indicate the "deadheading" travel, i.e., travel to be done while not sweeping.

Paths added:
node 4 to node 16
node 7 to node 2
node 14 to node 16

Another Example

Random Network
(seed= 433760)



Node Degrees

node #	1	2	3	4	5	6	7	8	9	10	11	12
in-degree	1	2	2	1	1	2	1	1	1	1	2	1
out-degree	3	1	1	1	1	1	2	1	1	1	2	1
polarity	-2	1	1	0	0	1	-1	0	0	0	0	0

Distances (edge lengths)

from \ to	1	2	3	4	5	6	7	8	9	10	11	12
1	0	13	54	999	999	999	999	999	61	999	999	999
2	13	0	999	999	999	999	999	999	999	999	999	999
3	999	44	0	999	999	999	999	999	999	999	999	999
4	999	999	18	0	999	999	999	999	999	999	999	999
5	999	999	999	12	0	999	999	999	999	999	999	999
6	999	999	999	999	7	0	999	999	999	999	999	999
7	999	999	999	999	999	999	0	999	999	999	46	61
8	999	999	999	999	999	999	23	0	999	999	999	999
9	999	999	999	999	999	999	999	7	0	999	999	999
10	999	999	999	999	999	32	999	999	999	0	999	999
11	999	999	999	999	999	26	999	999	999	55	0	999
12	999	999	999	999	999	999	999	999	999	999	28	0

999 <=> infinity, i.e., absence of arc

Path Lengths

from \ to	1	2	3	4	5	6	7	8	9	10	11	12
1	0	13	54	182	170	163	91	68	61	192	137	152
2	13	0	67	195	183	176	104	81	74	205	150	165
3	57	44	0	239	227	220	148	125	118	249	194	209
4	75	62	18	0	245	238	166	143	136	267	212	227
5	87	74	30	12	0	250	178	155	148	279	224	239
6	94	81	37	19	7	0	185	162	155	286	231	246
7	166	153	109	91	79	72	0	234	227	101	46	61
8	189	176	132	114	102	95	23	0	250	124	69	84
9	196	183	139	121	109	102	30	7	0	131	76	91
10	126	113	69	51	39	32	217	194	187	0	263	278
11	120	107	63	45	33	26	211	188	181	55	0	272
12	148	135	91	73	61	54	239	216	209	83	28	0

Predecessors

from \ to	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	1	5	6	11	8	9	1	11	7	7
2	2	0	1	5	6	11	8	9	1	11	7	7
3	2	3	0	5	6	11	8	9	1	11	7	7
4	2	3	4	0	6	11	8	9	1	11	7	7
5	2	3	4	5	0	11	8	9	1	11	7	7
6	2	3	4	5	6	0	8	9	1	11	7	7
7	2	3	4	5	6	11	0	9	1	11	7	7
8	2	3	4	5	6	11	8	0	1	11	7	7
9	2	3	4	5	6	11	8	9	0	11	7	7
10	2	3	4	5	6	10	8	9	1	0	7	7
11	2	3	4	5	6	11	8	9	1	11	0	7
12	2	3	4	5	6	11	8	9	1	11	12	0

Chinese Postman Problem in a Digraph

i:	1	2	3	6	7
Polarity:	⁻2	1	1	1	⁻1

Shortest Paths

from \ to	1	7
2	13	104
3	57	148
6	94	185

Solving transportation problem

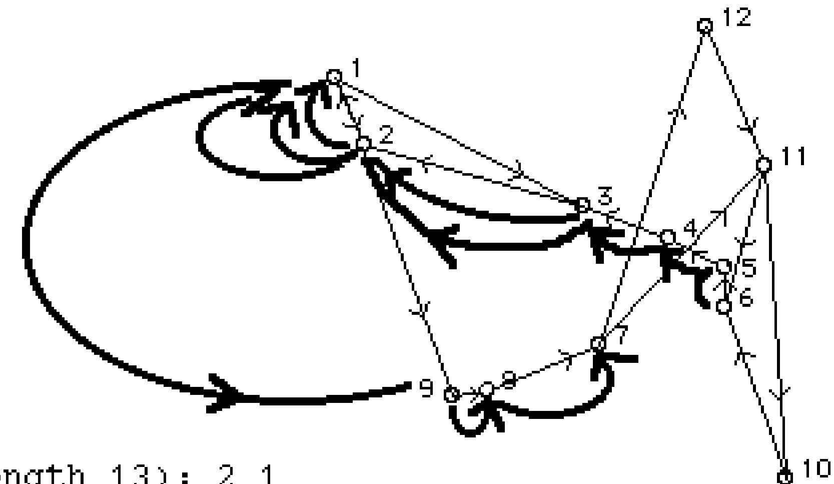
Solution of the transportation problem:

Paths added

from:	2	3	6
to:	1	1	7

1 × path (length 13): 2 1
1 × path (length 57): 3 2 1
1 × path (length 185): 6 5 4 3 2 1 9 8 7

Total length of paths added: 255



1 × path (length 13): 2 1
1 × path (length 57): 3 2 1
1 × path (length 185): 6 5 4 3 2 1 9 8 7

Total length of paths added: 255

Summary

- Solving the Postman Problem in an **undirected** network requires finding an optimal matching of the odd-degree nodes into pairs.
- Solving the Postman Problem in a **directed** network requires the solution of a transportation problem, with positive-polarity nodes as "sources" and negative-polarity nodes as "destinations".

In either case, the "cost" of a match or a shipment is the length of the shortest path between the two nodes.

REFERENCE

H.A. Eiselt, Michel Gendreau, & Gilbert Laporte,

"Arc Routing Problems, Part I: The Chinese Postman Problem", Operations Research, volume 43 (March/April 1995), pp. 231-242.

"Arc Routing Problems, Part II: The Chinese Postman Problem", Operations Research, volume 43 (May/June 1995), pp. 399-414.