

Answers

The sheet with its answers, **in blue**. Question 1(a) has many right answers and one of them is drawn; every other question has one.

Part 1 — Seven highways

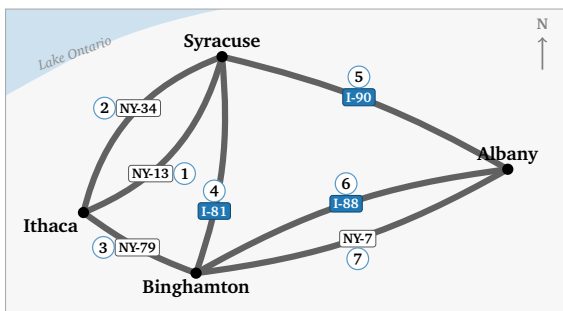
Seven highways join four Upstate New York cities: Ithaca, Syracuse, Binghamton, Albany.

Question 1(a): Drive every highway exactly once, without lifting your pencil. Start and finish anywhere; cities may repeat, highways may not. Write in each circle when you drive that road — 1 first, then 2, on to 7. Two are filled in: you start at Ithaca, take NY-13, then NY-34 straight back.

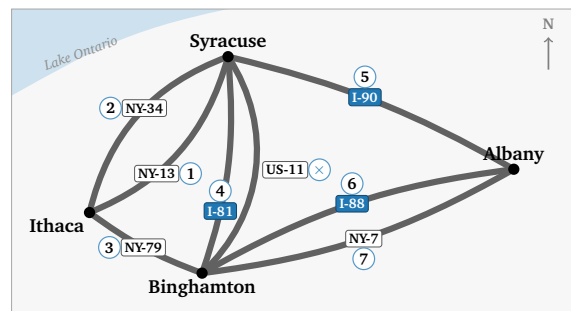
⇒ One drive out of many. Whichever roads it takes in between, it has to start at **Ithaca** and finish at **Albany**: those are the two cities with an odd number of roads, and an odd city is where a drive begins or ends.

Question 1(b): The same four cities, and an eighth highway: US-11, which really does run alongside I-81. Number your route again. Can you drive every highway without lifting your pencil?

⇒ **No**. Seven of the eight can be driven — the numbers above still work — and **US-11 is left over**, or I-81 if you would rather leave that one. US-11 pushes Syracuse to five roads and Binghamton to five, so **four** cities now have a left-over road, and a drive has one start and one finish.

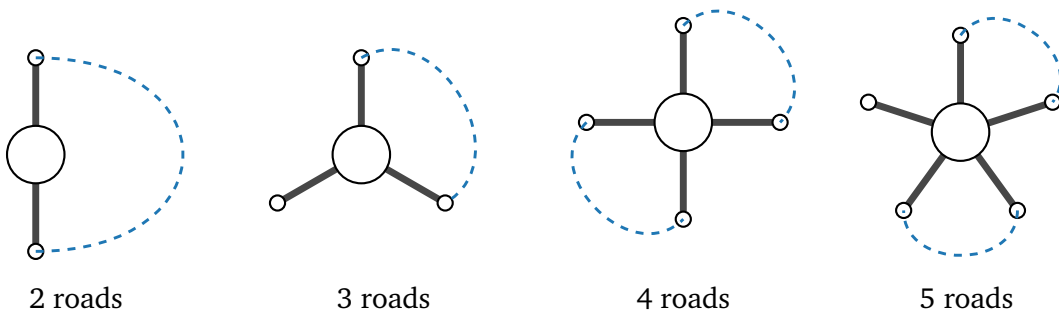


1(a): seven highways



1(b): with US-11

Question 2: One city, on its own. Arriving uses up one highway, leaving uses up another — so highways get used in **pairs**. Join two road-ends with a curved line to mean “in here, out there”. Pair up as far as you can.



	2 roads	3 roads	4 roads	5 roads
pairs you drew	1	1	2	2
roads left over	0	1	0	1

(a) Which of these cities can you drive straight *through*? ⇒ **2 and 4**, the ones with nothing left over.
 (b) What do those numbers share? ⇒ **They are even**.
 (c) A left-over road has no partner. What must this city be for the tour? ⇒ **Start or finish**.

Question 3: Count the highways at each city.

	I	S	B	A	how many have a left-over?
seven roads	3	4	4	3	2
with US-11 too	3	5	5	3	4

One start, one finish, so at most how many cities may have a left-over road? $\Rightarrow$ 2.

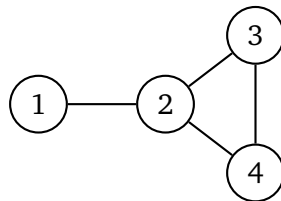
For the seven roads: can you drive every highway exactly once? If so, where does that drive begin and end? $\Rightarrow$ **Yes** — the drive existing means its two odd cities are its ends, so it runs **Ithaca** to **Albany**.

For the eight roads, with US-11: can you drive every highway exactly once? $\Rightarrow$ **No**.

Why is there no route with eight? $\Rightarrow$ **Four cities have a left-over road, and a drive has only one start and one finish.**

Part 2 — Three kinds of route

A different network: four stops, numbered 1 to 4.



Question 4(a): Fill in the table. The last column takes one of these three names.

Repeats nothing: **path**. Never repeats a line: **trail**. Repeats both: **walk**.

route	repeats a line?	repeats a stop?	name
1 2 3 2 4	yes (2–3 twice)	yes (2)	walk
1 2 3 4 2	no	yes (2)	trail
1 2 3 4	no	no	path

Question 4(b): Write a route from 1 to 2 that repeats a stop but no line.

$\Rightarrow$ **1 2 3 4 2**. The lines 1–2, 2–3, 3–4, 4–2 are four different lines; 2 is the stop that comes round twice.

Question 4(c): Your drive in 1(a) used no road twice, but it did come back to a city it had already visited. Which is it — a path, a trail, or a walk?

$\Rightarrow$ **A trail**. No road twice makes it a trail; coming back to a city it had already visited is what stops it being a path.

Question 4(d): It could not have been a path. How many different cities would a path of 7 roads have to touch? How many does the highway map have? Why does that rule out a path?

$\Rightarrow$ A path of 7 roads touches **8** cities — one more city than roads, because a path never comes back — and the map has only **4**. With nowhere near 8 cities to visit, the drive cannot be a path.

Part 3 — Predictions

Part 4 hands the student a notebook — a computer page that runs the network from Part 2. Worked out on paper first, then checked by the notebook.

Question 5(a): Write **1** in row i , column j if a line joins i and j , else **0**, for the network from Part 2.
5(b): Add up each row.

$A =$	1	2	3	4	row total
1	0	1	0	0	1
2	1	0	1	1	3
3	0	1	0	1	2
4	0	1	1	0	2

Each row total counts the **degree** of that node.

Question 5(c): A **2-step route** goes along one line, then along one more. Count them on the network from Part 2.

$\Rightarrow$ From 1 to 3: **1** of them, $1 \rightarrow 2 \rightarrow 3$. From 1 to 2: **0**, because **the first step has to go to 2, and the second step has to leave it again**. From node 2 back to node 2: **3**, out and back along each of its three lines.

Question 5(d): The notebook will square your grid. Say what you expect to find in A^2 .

$\Rightarrow$ Row 1, column 3: **1**. Row 1, column 2: **0**. The diagonal is **the degrees** — 1, 3, 2, 2 — because a 2-step route that comes back is one line out and the same line home.

$$A^2 = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 3 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}$$

Part 4 — The lab

The seven roads, read off the map — the order does not matter, but both Ithaca–Syracuse roads must be there and both Binghamton–Albany roads must be there:

```
ROADS = [(0, 1), (0, 1), (0, 2), (1, 2), (1, 3), (2, 3), (2, 3)]
```

```
def to_adjacency(edges, n):
    A = np.zeros((n, n), dtype=int)
    for i, j in edges:
        A[i, j] += 1
        A[j, i] += 1
    return A

def degrees(A):
    return A.sum(axis=1)

def euler_status(A):
    if not is_connected(A):
        return "impossible"
    odd = int(np.sum(degrees(A) % 2 == 1))
    if odd == 0:
        return "circuit"
    if odd == 2:
        return "path"
    return "impossible"
```

The verdicts: seven roads **path**, with US-11 **impossible** — 1(a) and 1(b) again, out of a rule that never looked at the picture.

Finished early. Every degree even and still no tour means the map is **in more than one piece**: two triangles, say,

```
CHALLENGE = [(0, 1), (1, 2), (2, 0), (3, 4), (4, 5), (5, 3)]
```

Every place has two roads, and no drive gets from one triangle to the other. A rule that only counted parity says circuit here and is wrong; that is what `is_connected(A)` is for. One road joining the two triangles makes its two ends odd and the answer path; a second road *beside the first*, between the same two places, makes everything even again and the answer circuit.

The whole notebook with every blank filled in is `lab-solutions.py`, built from `lab.py` by `tools/build_lab_notebooks.py m01-euler-tour`.