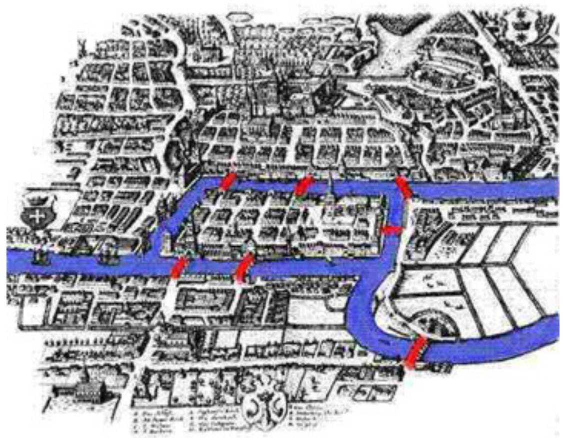
CSCI 3110 Lecture notes
Graph (3)

- **Euler Circuit**
  - **The Konigsberg Bridge problem:** Is it possible to tour through the city of Konigsberg without crossing any of the 7 bridges twice?



  - **Definition:** a cycle in a graph that visit every edge in the graph exactly once.
  - A graph that contains a Euler circuit is called a **Eulerian graph.**
  - **Requirement** for a Euler graph, degree of each vertex in the graph should be an even number

- **Hamiltonian Circle:**
  - **Definition:** A circuit in a graph that visits every vertex of the graph exactly one time
  - A graph that contains a Hamiltonian circuit is called a **Hamiltonian graph.**
  - Hemiltonian circle is NP-complete problem.

- **Polynomial problem, NP problem, NP complete problem**

The class P problems (Polynomial time problem) is the class of decision problems that are solvable by algorithms that run in polynomial time (as a function of input size). Tractable problems.
Problems that are solvable by algorithms that run in any of the following (worst-case) runtimes are in P: 1/n 1 lg n n n lg n $n^{k.}$
A problem that requires runtimes more than n^k for all k, are not in P.

Intractable problem
HALT is not in P; in fact HALT is unsolvable, period. (HALT is the problem of deciding whether, for an input program p that takes no data, p eventually halts. Turing showed this is unsolvable: no algorithm can correctly produce the right answer (yes or no) for every input program p.)

What is in between P and unsolvable problems? NP problems.

The Class NP (non-deterministic polynomial time**) problems** are problems that, although they may be hard to solve by algorithms that run fast, they can be **verified** quickly. :

NP = {q: q is a decision problem and there is a verification algorithm V for q, that runs in polynomial time as a function of the size n of the input i to q}

Class of NP Complete problems
Cook showed in 1971 that there is a special set of problems contains special problems, known as NP-complete problems, that are maximally difficult within NP. That is, an NP-complete problem p is one such that, roughly, if Ap solves p, then every q in NP can be solved by a variation on Ap that runs essentially as fast as Ap.
More precisely, p is NP-complete if:

1. p is in NP
2. all other problems q in NP can be "reduced in polynomial time" to solving p: there is a function f computable in polynomial-time such that for all inputs i to p, the correct answer to q(i) is yes iff the correct answer to p(f(i)) is yes.