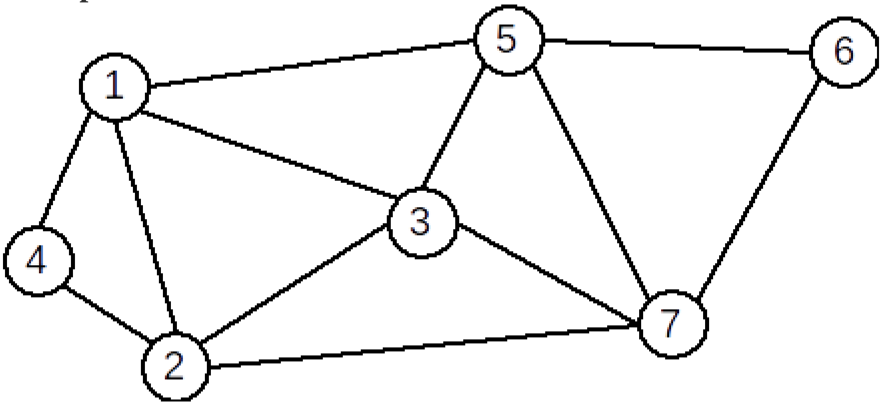
Graf hamiltonian

Definiție: Se numește **graf hamiltonian** un graf care conține un ciclu hamiltonian. Se numește **ciclu hamiltonian** un ciclu elementar care conține toate vârfurile grafului.

Exemplu: Graful următor este hamiltonian. Un ciclu hamiltonian este: [1,4,2,3,7,6,5,1]



Se numește **lanţ hamiltonian** un lanţ elementar care conține toate vârfurile grafului.

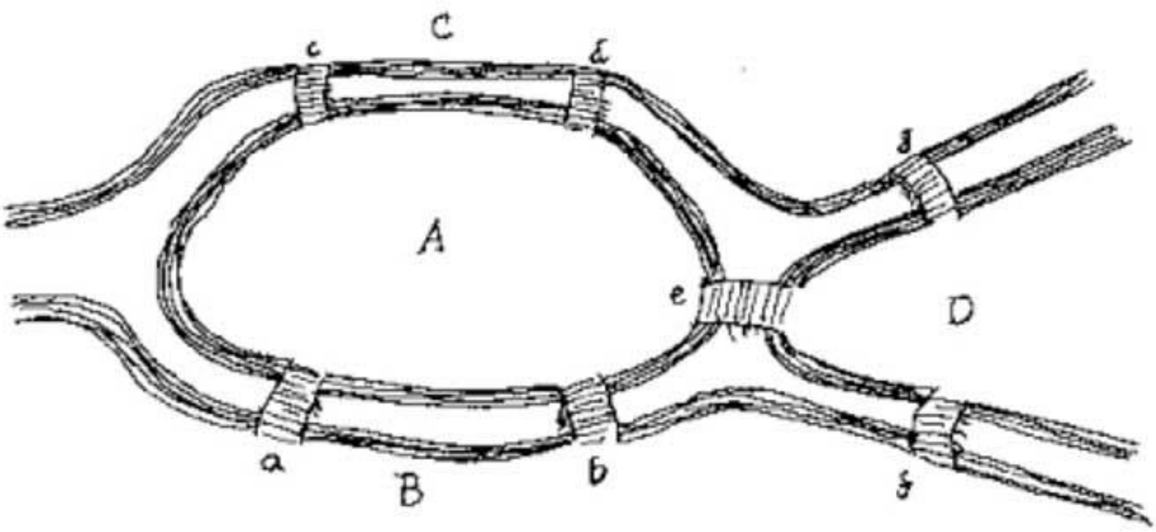
Teoremă: Fie G un graf neorientat. Dacă are n≥3 vârfuri şi gradul oricărui vârf verifică inegalitatea d(x)≥n/2 atunci G este hamiltonian. Reciproca nu este neapărat adevărată. (deci este o condiţie suficientă, nu şi necesară)

Propoziţie. Orice graf complet este graf hamiltonian.

Graf eulerian

https://www.pbinfo.ro/articole/5792/graf-eulerian

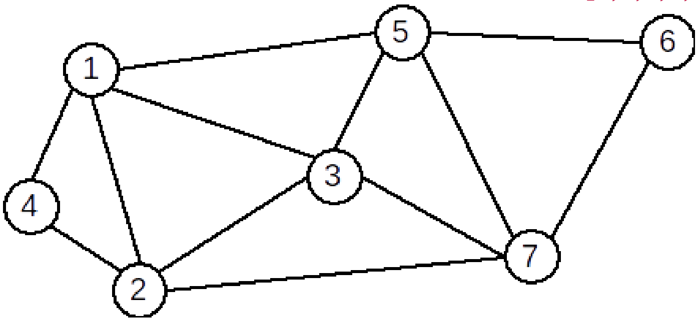
În 1736, matematicianul elvețian Leonard Euler a rezolvat o problemă cunoscută astăzi ca "Problema celor șapte poduri din Königsberg". Prin orașul Königsberg, azi Kaliningrad trece un râu pe care sunt două insule, acestea și malurile fiind unite prin poduri ca în imaginea de mai jos.



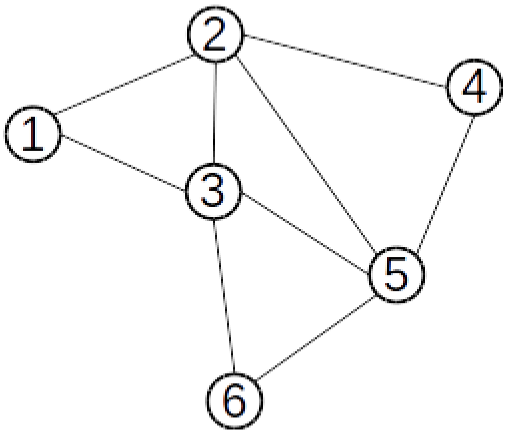
Leonard Euler a demonstrat că nu este posibil ca o persoană să traverseze fiecare pod o singură dată. În onoarea sa, o categorie specială de grafuri se numesc **grafuri euleriene**.

Definiție: Se numește **graf eulerian** un graf care conține un ciclu eulerian. Se numește **ciclu eulerian** un ciclu care conține toate muchiile grafului.

Exemplu 1: Graful următor este eulerian. Un ciclu eulerian este: [1,4,2,1,3,2,7,3,5,7,6,5,1]



Exemplu 2: În graful de mai jos ciclul (1 2 4 5 3 6 5 2 3 1) este eulerian.



Teoremă de caracterizare: Un graf G = (X,U), fără vârfuri izolate, este eulerian dacă şi numai dacă este conex şi gradele tuturor vârfurilor sale sunt numere pare.

Se numește **lanț eulerian** un lanț simplu care conține toate muchiile grafului (fiind lanț simplu, fiecare muchie apare o singură dată).

Pentru determinarea unui ciclu eulerian se pot folos mai mulți algoritmi. Unul dintre aceştia este asemănător cu parcurgerea în adâncime.

Orice graf eulerian poate fi descompus ca reuniune de cicluri disjuncte. Astfel, algoritmul formeaza un ciclu oarecare, parcurgandu-l apoi in ordine inversa pentru a intercala recursiv ciclurile formate de muchiile ramase in graf.

Vom prezenta succint algoritmul in pseudocod:

```
euler (nod v)  //antet
cat timp (v are vecini)
w = un vecin aleator al lui v
sterge_muchie (v, w)
euler (w)
sfarsit cat timp
adauga v la ciclu
```

Important:

- la vizitarea unui vârf, acesta nu va fi marcat, pentru a permite revenirea în el;
- finitudinea algoritmului este asigurată de faptul că muchiile se elimină din graf;

Considerăm un graf cu n vârfuri, memorat prin intermediul matricei de adiacență. Vectorul L reprezintă lista în care se memorează nodurile ciclului eulerian. Toate variabilele sunt globale:

```
void Euler(int k)
{
   for(int i = 1 ; i <= n ; i ++)
      if(A[k][i] == 1)
      {
         A[k][i] = A[i][k] = 0;
         Euler(i);
      }
   L[++p] = k;
}
...
Euler(1);
// se afiseaza vectorul L cu p elemente
...
```

Algoritmul de mai sus poate fi utilizat și pentru determinarea unui lanț eulerian. Parcurgerea trebuie să înceapă însă din unul dintre vârfurile cu grad impar.