DISKRETNA MATEMATIKA: OJLEROV GRAF

SADRŽAJ:

- Istorijski osvrt na 7 mostova Kenigsberga
- Definicija Ojlerovog i polu Ojlerovog grafa
- Uslov da graf bude Ojlerov
- Uslov da graf bude polu Ojlerov
- Podsećanje Priferovog koda
- Podsećanje težinskog grafa
- ...

Ojlerov graf

- Ako je graf moguće nacrtati u **jednom potezu**, **tako da ne dižemo olovku sa papira, a da se na kraju vratimo u tačku iz koje smo krenuli**, kažemo da je **graf Ojlerov**.

- Najpoznatiji problem koji se odnosi na ispitivanje postojanja takve konture u grafu, a smatra se ujedno i začetkom teorije grafova, jeste problem *sedam mostova Kenigsberga.*

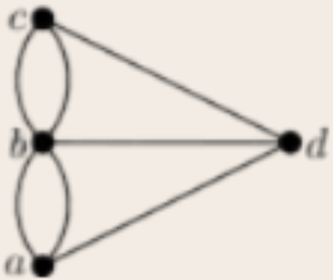
Sedam mostova Kenigsberga

Sedam mostova Kenigsberga (Kalinjingrada). Problem je rešio 1735. godine švajcarski matematičar **Leonard Ojler**. Kenigsberg (*danas Kaliljingrad u Rusiji*) je bio grad u Pruskoj, kroz koji prolazi reka Pregolja. U jednom delu grada reka obilazi dva ostrva. Tokom izgradnje grada, izgrađeno je ukupno **sedam mostova koja povezuju ostrva sa levom i desnom obalom reke**, ali i jedno sa drugim. Problem je bio odrediti šetnju po gradu, tako da se svaki most pređe tačno jednom i da se vratimo odakle smo krenuli.

Leonard Ojler

Sedam mostova Kenigsberga

Ključni doprinos u rešavanju ovog problema predstavljala je **grafovska reprezentacija datog problema**.
Ojler je problem dalje rešio **razmatranjem parnosti broja grana koje izlaze iz pojedinačnih čvorova**. Tako je konstatovao da je nemoguće napraviti traženu šetnju kroz graf.



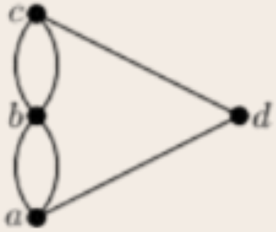
Leonard Ojler

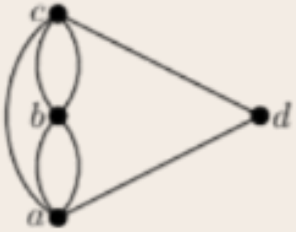
Ojlerov graf

Neka je G graf bez petlji. **Ojlerov put** je staza u grafu koja sadrži sve grane tog grafa. **Ojlerova tura** je Ojlerova staza u kojoj su početni i krajnji čvor jednaki. Znači, za šetnju u grafu ćemo reći da je Ojlerov put, ako zadovoljava sledeća dva uslova:

(i) *ne postoje dve jednake grane u šetnju;*

(ii) *(svaka grana grafa se pojavljuje u šetnji.*



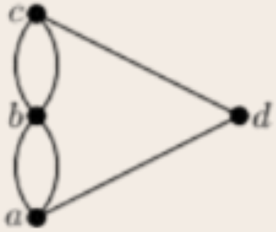


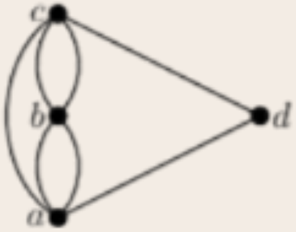
Primer jednog Ojlerovog puta u G1 je ***adcbabc****, dok je primer Ojlerove ture u G2 je staza* ***adcbabca***

Ojlerov graf

Graf je **Ojlerov** ako je povezan graf koji sadrži Ojlerovu turu.

Graf je **polu Ojlerov** ako je povezan i sadrži Ojlerov put.





*Primer jednog Ojlerovog puta u G1 je **adcbabc**, dok je primer Ojlerove ture u G2 je staza **adcbabca***

Teorema: *„Neka je G graf. Graf G je Ojlerov ako i samo ako je povezan i svaki čvor u G je parnog stepena."*

Potreban i dovoljan uslov za postojanje Ojlerovog grafa

Dokaz:

($\longrightarrow$) Graf je povezan po definiciji Ojlerovog grafa. Neka je

$$u_1 e_1 u_2 e_2 \ldots u_n e_n u_1$$

Ojlerova tura u G. Posmatrajmo sada proizvoljan čvor v ∈ V(G). Ako se čvor v pojavljuje l puta u konturi u slučaju kada je $v \neq u_1$, odnosno $l + 1$ ako je $v = u_1$, onda je stepen tog čvora $d_G(v)=2l$.

($\longleftarrow$) Posmatrajmo u G stazu najveće dužine:

$$u_1 e_1 u_2 e_2 \ldots u_n e_n u_{n+1}$$

Pokazaćemo da su prvi i poslednji čvor isti, kao i da se svi čvorovi i grane grafa pojavljuju u toj stazi.

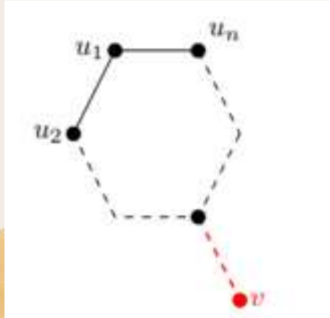
Dokaz:

1) $u_1 = u_{n+1}$:

Ako pretpostavimo suprotno, da je u_1 različito od u_{n+1}, onda je u toj konturi neparan broj grana incidentan sa u_1 u toj stazi. Kako je stepen čvora u_1 paran, postoji grana e iz skupa E(G) koja nije sadržana u posmatranoj stazi. U tom slučaju bismo mogli kreirati dužu stazu od posmatrane, što dovodi do kontradikcije.

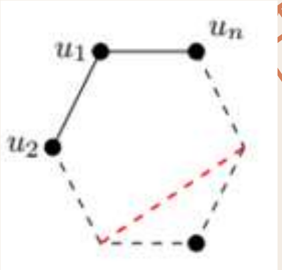
2) $\{u_1, \ldots, u_n\} = V(G)$:

Pretpostavimo suprotno, da postoje čvorovi koji nisu na posmatranoj stazi. Kako je G povezan, postoji grana $\{u_i, v\}$, $i \in \{1, \ldots, n\}$, sa osobinom $v \notin \{u_1, \ldots, u_n\}$. U tom slučaju možemo konstruisati stazu veće dužine od posmatrane.



3) $\{e_1, \ldots, e_n\} = E(G)$:

Sada kada znamo da je posmatrana staza u stvari kontura koja sadrži sve čvorove grafa, pokazaćemo da su sve grane grafa na toj konturi. Ako pretpostavimo suprotno, da postoji grana koja nije na toj konturi, onda bismo ponovo mogli konstruisati dužu stazu od posmatrane.



Teorema: *„Neka je G = (V,E) graf koji nije Ojlerov. Graf G je polu Ojlerov ako i samo ako je G povezan i ima tačno dva čvora neparnog stepena.“*

Kako je svaki Ojlerov graf isovremeno i polu Ojlerov graf, u sledeće tvrđenje izdvaja potreba i dovoljan uslov za postojanje Ojelrovog puta u grafu koji nema Ojlerovu turu.

Dokaz:

(→) Sličnim rezonovanjem kao u prethodnom dokazu, za svaki čvor grafa koji nije na krajevim Ojlerovog puta, na tom putu se pojavljuje paran broj grana koje su incidentne sa tim čvorom. Za dva čvora na kraju puta imamo, osim eventualnog parnog broja grana unutar staze, još po jednu granu koja im je incidentna, odakle dobijamo da ta dva čvora imaju neparne stepene.

(←) Neka su u i v jedini čvorovi grafa neparnog stepena. Posmatrajmo sada graf G' koji dobijamo od grafa G dodavanjem nove grane e koja je incidentna sa čvorovima u i v (ona može biti paralelna nekim već postojećime granama). U grafu G' su sada svi čvorovi parnog stepena. Prema prethodnoj teoremi, G' sadrži Ojleorvu turu. Po definiciji, ta tura sadrži granu G. Oduzimanjem grane e iz Ojlerove ture grafa G' dobijamo Ojlerov put u grafu G.

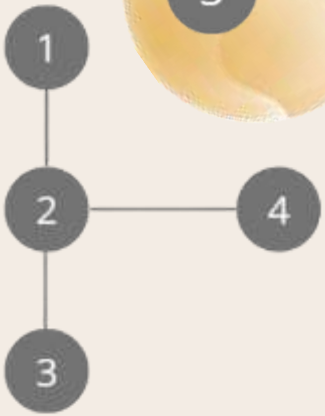
Primer – Priferov kod odredjuje stablo

Priferov niz je (2,2,4,5)
n = 5 + 2 = 6 – broj čvorova
Čvorovi su {1,2,3,4,5,6}
Broj pojavljivanja čvorova:
Čvorovi 4 i 5 – 1 put; Čvor 2 – 2 puta;
Čvorovi 1,3 i 6 – 0 puta;

1) Dodaje se grana (1,2)
Priferov kod: (2,4,5)

2) Dodaje se grana (3,2)
Priferov kod: (4,5)

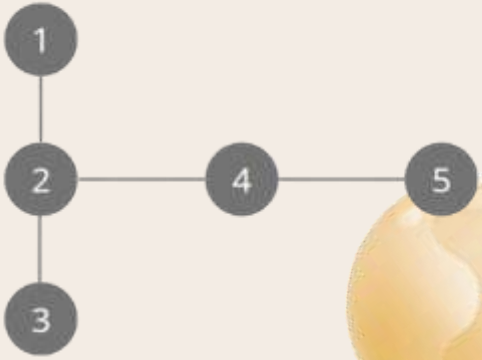
3) Dodaje se grana (2,4)
Priferov kod: 5



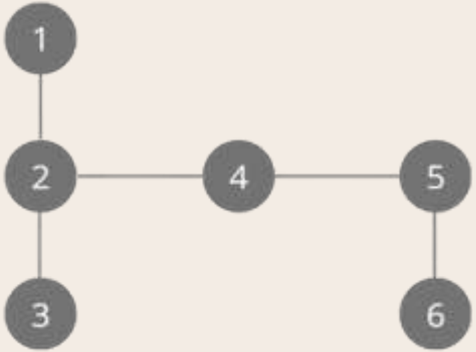
Primer – Priferov kod odredjuje stablo

Priferov niz je (2,2,4,5)
n = 5 + 2 = 6 – broj čvorova
Čvorovi su {1,2,3,4,5,6}
Broj pojavljivanja čvorova:
Čvorovi 4 i 5 – 1 put; Čvor 2 – 2 puta;
Čvorovi 1,3 i 6 – 0 puta;

4) Dodaje se grana (4,5)



5) Dodaje se grana (5,6)



Algoritam za odredjivanje Ojlerove ture u grafu

```java
public List<Integer> findEulerTour() {
    // Lista u kojoj ćemo čuvati Ojlerovu turu
    List<Integer> eulerTour = new ArrayList<>();

    // Koristimo stack (LIFO - last in, first out) za praćenje putanje kroz graf
    Stack<Integer> stack = new Stack<>();

    // Početni čvor za Ojlerovu turu je čvor sa indeksom 0
    stack.push(0);

    // Petlja se izvršava dokle god stack nije prazan
    while (!stack.isEmpty()) {
        // Trenutni čvor (vrh stoga) koji trenutno obrađujemo
        int node = stack.peek();

        // Ako trenutni čvor ima susede (tj. još nije prošao kroz sve svoje grane)
        if (!adj[node].isEmpty()) {
            // Uzimamo prvog suseda iz liste suseda (uklanjamo granu između trenutnog čvora i suseda)
            int neighbor = adj[node].remove(0);

            // Takođe uklanjamo trenutni čvor iz liste suseda njegovog suseda
            adj[neighbor].remove((Integer) node);

            // Gurnemo suseda na stog, jer ćemo sada preći na njega
            stack.push(neighbor);
        } else {
            // Ako trenutni čvor nema više suseda (sve grane su prošle), dodajemo ga u Ojlerovu turu
            eulerTour.add(stack.pop());
        }
    }

    // Vraćamo listu koja predstavlja Ojlerovu turu
    return eulerTour;
}
```

```java
public static void main(String[] args) {
    OjlerovGraf graph = new OjlerovGraf(5);
    graph.addEdge(0, 1);
    graph.addEdge(0, 2);
    graph.addEdge(1, 2);
    graph.addEdge(1, 3);
    graph.addEdge(3, 4);
    graph.addEdge(4, 0);

    List<Integer> eulerTour = graph.findEulerTour();
    System.out.println("Ojlerova tura: " + eulerTour);
}
```

Hvala na pažnji!