Ojlerov graf

04.02.2025.

1 Ojlerov graf

1.1 Sedam mostova Kenigsberga

Problem Kenigsbergovih mostova, zabavna matematička zagonetka smeštena u starom pruskom gradu Kenigsbergu (danas Kalinjingrad, Rusija), doveo je do razvoja grana matematike poznatih kao topologija i teorija grafova. U početku 18. veka, građani Kenigsberga provodili su dane hodajući po složenoj mreži mostova koji su prelazili vodu reke Pregel (Pregolja), koja je okruživala dva centralna kopna povezana mostom (3). Prvo kopno (ostrvo) bilo je povezano sa donjom obalom Pregela sa dva mosta (5 i 6), dok je drugo kopno, koje je podelilo Pregel na dva kraka, bilo povezano sa donjom obalom jednim mostom (7) i sa gornjom obalom jednim mostom (4), ukupno sedam mostova. Prema narodnoj priči, postavljeno je pitanje da li bi građanin mogao da prođe kroz grad na način da svaki most pređe tačno jednom.

1735. godine švajcarski matematičar Leonard Ojler dao je rešenje ovog problema, zaključivši da takav hod nije moguć. Da bi to potvrdio, pretpostavimo da takav hod postoji. U jednom susretu sa određenim kopnom, osim početnog ili terminalnog, moraju se preći dva različita mosta: jedan za ulazak u kopno i jedan za izlazak iz njega. Dakle, svako takvo kopno mora biti krajnja tačka broja mostova koji je dvostruko veći od broja puta kada se to kopno sretne tokom hoda. Stoga svako kopno, uz mogući izuzetak početnih i krajnjih, mora biti krajnja tačka parnog broja mostova. Međutim, za kopna Kenigsberga, A je krajnja tačka pet mostova, a B, C i D su krajnje tačke tri mosta. Hod je stoga nemoguć.

Tek skoro 150 godina kasnije matematičari su počeli da posmatraju problem Kenigsbergovih mostova kao graf koji se sastoji od čvorova (temena) koja predstavljaju kopna i lukova (grana) koji predstavljaju mostove. Stepen temena grafa određuje broj grana koje su povezane sa tim temenom. U savremenoj teoriji grafova, Ojlerov put prelazi svaku granu grafa jednom i samo jednom. Tako je Ojlerovo tvrđenje da graf koji poseduju takav put ima najviše dva temena neparnog stepena bila prva teorema u teoriji grafova.

1.2 Ojlerov graf i polugraf

Definicija 1. Ojlerov put (ili Ojlerova staza) je staza u grafu koja prolazi kroz svaku granu tačno jednom. **Ojlerova tura** je Ojlerova staza u kojoj je početni i krajnji čvor jednak.

Primer: Koji od neusmerenih grafova na slici imaju Ojlerovu turu? Od onih koji nemaju, koji imaju Ojlerovu stazu?

Rešenje: Graf G_1 ima Ojlerovu turu, na primer: a, e, c, d, e, b, a. Graf G_2 nema ni Ojlerovu turu ni Ojlerov put. Graf G_3 nema Ojlerovu turu, međutim, ima Ojlerov put, naime: a, c, d, e, b, d, a, b.

Definicija 2. Graf je Ojlerov ako sadrži Ojlerovu turu. Graf je polu Ojlerov ako sadrži Ojlerov put.

1.2.1 Potrebni i dovoljni uslovi za Ojlerovu turu i putanju

Teorema 1. Graf G je Ojlerov ako i samo ako je povezan i svaki čvor u G je parnog stepena.

Dokaz:

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

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

Ojlerova tura u G. Posmatrajmo sada proizvoljan čvor $v \in V(G)$. Ako se čvor v pojavljuje l puta u konturi u slučaju kada je $v \neq u_1$, odnosno $l+1$ ako je $v = u_1$, tada važi

$$d_G(v) = 2l.$$

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

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

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

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

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

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

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

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.

Sada možemo rešiti problem Kenigsbergovih mostova. Pošto multigraf koji predstavlja ove mostove ima četiri temena neparnog stepena, on nema Ojlerovu turu. Ne postoji način da se krene sa određenog mesta, pređe svaki most tačno jednom i vrati se na početnu tačku.

Teorema 2. 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.

Dokaz:

($\Rightarrow$) Sličnim rezonovanjem kao u prethodnom dokazu, za svaki čvor grafa koji nije na krajevima 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.

($\Leftarrow$) Neka su u i v jedini čvorovi grafa neparnog stepena. Posmatramo 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 postojećim granama). U grafu G' su sada svi čvorovi parnog stepena. Prema Teoremi 1, G' sadrži Ojlerovu turu. Po definiciji, ta tura sadrži granu e. Oduzimanjem grane e iz Ojlerove ture grafa G' dobijamo Ojlerov put u grafu G.

Primer: Koji od neusmerenih grafova na slici ima Ojlerovu putanju?

Rešenje: G_1 sadrži tačno dva temena neparnog stepena, naime, b i d. Dakle, ima Ojlerov put koji mora imati b i d kao krajnje tačke. Jedan takav Ojlerov put je d, a, b, c, d, b. Slično tome, G_2 ima tačno dva temena neparnog stepena, naime, b i d. Dakle, ima Ojlerov put koji mora imati b i d kao krajnje tačke. Jedan takav Ojlerov put je b, a, g, f, e, d, c, g, b, c, f, d. Graf G_3 nema Ojlerov put jer ima šest temena neparnog stepena.

1.2.2 Algoritam za pronalaženje Ojlerove ture

Hierholzerov algoritam

1. Algoritam prvo proverava da li svi čvorovi imaju paran stepen. Ako neki čvor ima neparan stepen, Ojlerova tura ne postoji.

2. Počinjemo iz bilo kojeg čvora (jer svi imaju paran stepen).

3. Gradnja ture pomoću steka: Koristimo stek za čuvanje trenutnog puta kroz graf.

- Ako trenutni čvor ima dostupne grane, dodajemo ga na stek i prelazimo na jedan od njegovih suseda, uklanjajući granu između trenutnog i sledećeg čvora.
- Ako trenutni čvor više nema grana, dodajemo ga u rezultat i vraćamo se nazad stekom.

4. Dobijanje konačne ture: Kada obradimo sve grane, dobijamo Ojlerovu turu u ispravnom redosledu.

[ ]:
```
def ojlerova_tura(graf):
    """
    Pronalaženje Ojlerove ture koristeći Hierholzerov algoritam.

    Argument:
    graf -- rečnik gde je ključ čvor, a vrednost lista susednih čvorova.

    Povratna vrednost:
```

```python
    Lista čvorova u redosledu Ojlerove ture, ili poruka ako tura ne postoji.
    """
    # Provera uslova: svi čvorovi moraju imati paran stepen
    for cvor in graf:
        if len(graf[cvor]) % 2 != 0:
            return "Ojlerova tura ne postoji jer neki čvor ima neparan stepen."

    # Kopiramo graf da bismo manipulisali granama
    graf_kopija = {cvor: lista[:] for cvor, lista in graf.items()}

    stek = []  # Stack za praćenje trenutnog puta
    tura = []  # Lista za čuvanje Ojlerove ture

    # Početni čvor (bilo koji čvor može biti početni u Ojlerovoj turi)
    trenutni_cvor = next(iter(graf))

    # Glavni algoritam
    while stek or graf_kopija[trenutni_cvor]:
        if not graf_kopija[trenutni_cvor]:
            # Ako trenutni čvor nema više grana, dodajemo ga u turu
            tura.append(trenutni_cvor)
            trenutni_cvor = stek.pop()
        else:
            # Ako ima susede, prelazimo na sledeći čvor
            stek.append(trenutni_cvor)
            sledeci_cvor = graf_kopija[trenutni_cvor].pop()
            graf_kopija[sledeci_cvor].remove(trenutni_cvor)
            trenutni_cvor = sledeci_cvor

    tura.append(trenutni_cvor)  # Dodajemo poslednji čvor
    return tura

# Primer grafa
graf = {
    1: [2, 6],
    2: [1, 3, 6, 7],
    3: [2, 4, 5, 7],
    4: [3, 5],
    5: [3, 4, 6, 7],
    6: [1, 2, 5, 7],
    7: [2, 3, 5, 6]
}

# Poziv funkcije
rezultat = ojlerova_tura(graf)
print("Ojlerova tura:", rezultat)
```

```
Ojlerova tura: [1, 2, 3, 4, 5, 3, 7, 2, 6, 5, 7, 6, 1]
```

1.3 Priferov kod

Primer: Odredite stablo čiji je Priferov kod (1, 1, 1, 1, 6, 5).

Rešenje: Dati Priferov kod ima šest elemenata, pa će odgovarajuće stablo imati **6+2=8** čvorova.

Prvi broj u Priferovom kodu je 1 (**1**, 1, 1, 1, 6, 5), a najmanji broj koji nije uključen u kod je 2, pa povezujemo 1 i 2.

Sledeći broj u kodu je i dalje 1 (1, **1**, 1, 1, 6, 5), a najmanji broj koji nije uključen sada je 3, pa povezujemo 1 i 3.

Sledeći broj u kodu je i dalje 1 (1, 1, **1**, 1, 6, 5), a najmanji broj koji nije uključen sada je 4, pa povezujemo 1 i 4.

Sledeći broj u kodu je ponovo 1 (1, 1, 1, **1**, 6, 5), a najmanji broj koji nije uključen sada je 7, pa povezujemo 1 i 7.

Sledeći broj u kodu sada je 6 (1, 1, 1, 1, **6**, 5), a najmanji broj koji nije uključen je 1, pa povezujemo 6 i 1

Sada je broj u kodu 5 (1, 1, 1, 1, 6, **5**), a najmanji broj koji nije uključen je 6, pa povezujemo 5 i 6.

Na kraju, ostala su samo dva broja koji nisu u kodu: 5 i 8. Povezujemo ih.

I time smo završili konstruisanje stabla.

1.4 Težinski graf

Definicija. Težinski graf je graf $G = (V, E, w)$, gde je: - V skup čvorova, - E skup ivica, - $w : E \to \mathbb{R}$ funkcija koja dodeljuje težinu (realni broj) svakoj ivici $e \in E$.