Zadatak 7

Grafovi su matematičke strukture koje se sastoje od **čvorova** (ili **vrhova**) i **grana** koje ih povezuju. Oni predstavljaju širok spektar problema u nauci, tehnologiji, društvu i svakodnevnom životu, kao što su mreže puteva, društvene mreže, komunikacione mreže i mnoge druge.

Istorijski osvrt na grafove

Grafovi su formalno ušli u matematiku kroz rad švajcarskog matematičara **Leonarada Ojlera (Leonhard Euler)** 1736. godine. Njegov rad na **problemu sedam mostova Kenigsberga** (današnji Kalinjingrad, Rusija) se smatra početkom teorije grafova.

Problem sedam mostova Kenigsberga

Kenigsberg je bio grad kroz koji protiče reka Pregel. Grad je bio podeljen na četiri kopnena dela, povezana sa sedam mostova. Problem je glasio:

- **Da li je moguće proći kroz grad tako da se svaki most pređe tačno jednom?**

Ojler je dokazao da je to **nemoguće** i time je postavio temelje za analizu grafova. U svom rešenju, Ojler je apstrahovao problem tako što je grad predstavio kao **graf**:

- Svaki deo zemlje predstavljen je čvorom.
- Svaki most predstavljen je granom koja povezuje dva čvora.

Ojler je tada definisao **Eulerov put**:

- Put u grafu koji prolazi kroz svaku granu tačno jednom.
- Dokazao je da takav put postoji samo ako graf ima **najviše dva čvora sa neparnim stepenom** (brojem grana koje izlaze iz čvora).

Značaj Ojlerovog rada

Ojlerovo rešenje nije samo rešilo problem sedam mostova, već je uvelo koncept **teorije grafova**, koji se od tada koristi u različitim disciplinama.

Osnovni pojmovi u teoriji grafova

1. **Graf**: Sastoji se od skupa čvorova V i skupa grana E, formalno zapisan kao G=(V,E).
    - **Usmereni graf**: Grane imaju smer.
    - **Neusmereni graf**: Grane nemaju smer.
2. **Put**: Sekvenca čvorova u grafu gde su svaki uzastopni par čvorova povezani granom.

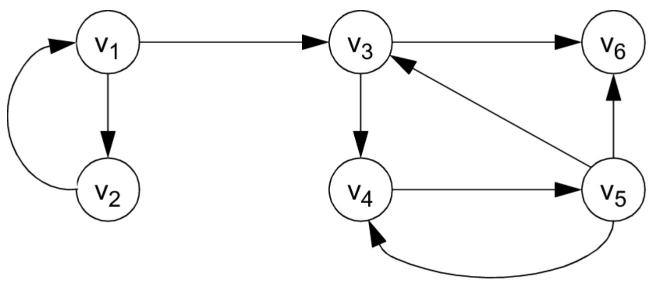
3. **Ciklus**: Put koji počinje i završava se u istom čvoru.

4. **Stepen čvora**: Broj grana koje izlaze iz čvora (za neusmereni graf) ili ulaze/izlaze (za usmereni graf).
5. **Povezanost**:
    - **Povezan graf**: Postoji put između svakog para čvorova.
    - **Nepovezan graf**: Postoje čvorovi između kojih ne postoji put.
6. **Posebni tipovi grafova**:
    - **Stablo**: Povezan graf bez ciklusa.
    - **Potpuni graf**: Svaki par čvorova je direktno povezan.

Usmeren i neusmeren multigraf

Usmereni multigraf (multifigraf) je uređena trojka G = (V, E, Ψ) gde je:
- $V \neq \varnothing$ konačan skup čvorova
- E je skup grana, pri čemu je $V \cap E = \varnothing$
- $\Psi : E \rightarrow \{ (u, v) : u, v \in V, u \neq v\}$ **funkcija incidencije**.

Uglavnom ćemo se baviti grafovima kojima grane nisu usmerene i u kojima nema paralelnih grana - grane e i e' sa osobinama Ψ(e) = Ψ(e'). Za takav graf kažemo da je prost.



slika 1. Usmereni multigraf

Neusmereni multigraf (multigraf) je uređena trojka G=(V,E,Ψ) gde je:

- $V \neq \varnothing$ konačan skup čvorova.
- E skup grana (ivica), pri čemu $V \cap E=\varnothing$ (čvorovi i grane su disjunktni skupovi).
- $\Psi : E \rightarrow \{ \{u, v\} : u, v \in V, u \neq v\}$ **funkcija incidencije**, koja svakoj ivici dodeljuje neuređeni par čvorova koje ta ivica spaja.
    - Ako u = v, tada se radi o **petlji** (ivica spaja čvor sa samim sobom).
    - Ako u ≠ v, tada se radi o običnoj ivici između čvorova u i v.

Oblasti primene teorije grafova

- **Računarske mreže i algoritmi**: Optimizacija internetskih protokola i rutiranja podataka (npr. Dijkstra algoritam, BFS, DFS).
- **Analiza društvenih mreža**: Proučavanje klika, zajednica i centralnosti u mrežama (npr. PageRank algoritam).

- **Transport i logistika**: Planiranje ruta u saobraćaju (GPS algoritmi), rešavanje problema trgovačkog putnika (TSP).
- **Bioinformatika i genetika**: Sekvenciranje DNK i analiza genetskih mreža (npr. otkrivanje veza između gena).
- **Molekularna hemija**: Modelovanje molekula i optimizacija lekova (čvorovi su atomi, a grane hemijske veze).
- **Elektroinženjering i energetske mreže**: Analiza električnih kola i projektovanje komunikacionih mreža.
- **Geografija i GIS**: Modelovanje puteva i pronalaženje najkraćih ruta (npr. A* algoritam).
- **Ekonomija i tržišne mreže**: Analiza tokova roba i odnosa između kompanija (npr. modelovanje finansijskih mreža).
- **Veštačka inteligencija**: Graf-neuronske mreže (GNN) za analizu podataka predstavljenih kao grafovi.

Definicija: Prost graf

Prosti graf $G = (V, E)$ je matematička struktura koja se sastoji od:

1. **Skupa čvorova (tačaka)** V, koji se nazivaju **temena** (vertices).
2. **Skupa grana** E, koje su **neusmereni parovi temena** iz V.

Grane povezuju temena, i svaka grana $e \in E$ može se zapisati kao $e = \{u, v\}$, gde $u, v \in V$.

U prostom grafu:

- Nema više grana između istih parova temena (nema duplih grana).
- Nema petlji, tj. grana koje povezuju temena sa samim sobom (npr. {v,v} nije dozvoljeno).

Formalna svojstva prostog grafa

- Skup temena V ima n=∣V∣ elemenata.
- Skup grana E ima m=∣E∣ elemenata.
- Maksimalan broj mogućih grana u prostom grafu sa n temena je $n(n - 1)/2$ (svaki par temena može biti povezan jednom granom).

Primer prostog grafa

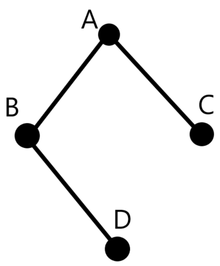
Razmotrimo prost graf G=(V,E), gde su:

- $V = \{A, B, C, D\}$ — skup temena, i
- $E = \{\{A, B\}, \{A, C\}, \{B, D\}\}$ — skup grana.

Ilustracija grafa

Možete zamisliti ili nacrtati sledeći graf:

- Čvorovi: A,B,C,D
- Grane: postoji veza između A i B, A i C, te B i D.



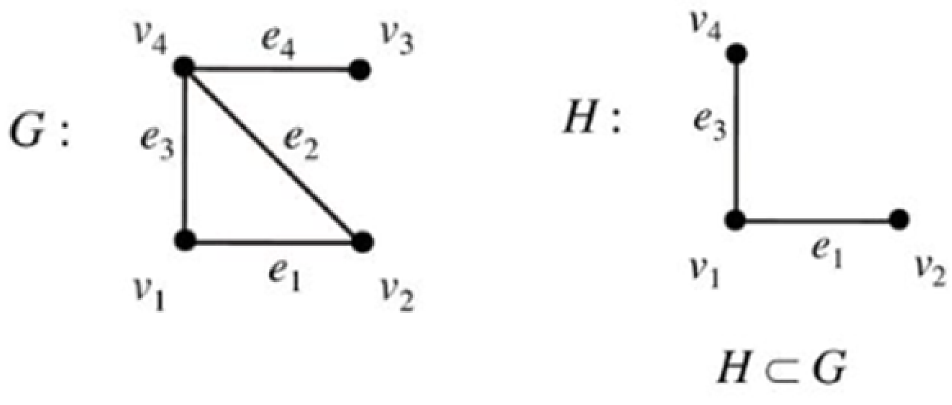
Analiza grafa:

- Broj temena: $|V|=4$.
- Broj grana: $|E|=3$.
- Ovo je prost graf jer:
    1. Nema petlji (npr. nema grane $\{A,A\}$).
    2. Svaki par temena ima najviše jednu vezu.

Definicija: Podgraf

Podgraf grafa $G=(V,E)$ je graf $G'=(V',E')$, gdje su:

- $V' \subseteq V$ skup čvorova podgrafa,
- $E' \subseteq E$ skup grana podgrafa, pri čemu svaka grana $e \in E'$ povezuje samo čvorove iz V'.

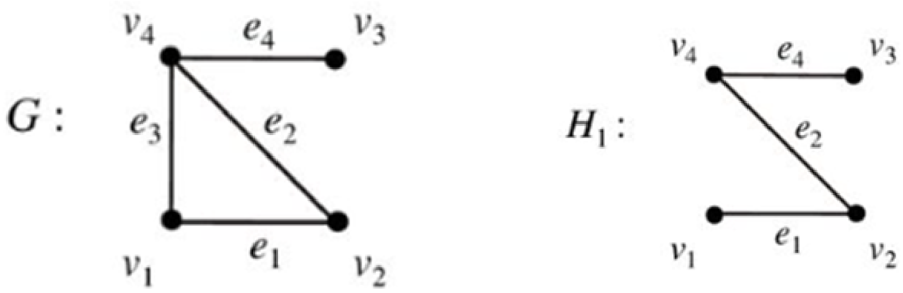


Definicija: Pokrivajući podgraf

Pokrivajući podgraf grafa $G=(V,E)$ je podgraf $G'=(V,E')$, gdje:

- $E' \subseteq E$ skup grana podgrafa,

- G′ sadrži sve čvorove V grafa G.
  Drugim riječima, pokrivajući podgraf ima isti skup čvorova kao originalni graf, ali može imati podskup njegovih grana.



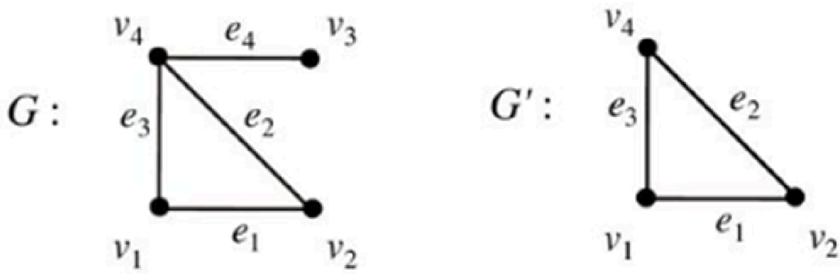
H_1 pokrivajući podgraf G

Definicija: Podgraf indukovan skupom čvorova

Podgraf grafa G=(V,E), indukovan skupom čvorova V′ ⊆ V, je graf G′=(V′,E′), gdje:

- V′ čvorovi podgrafa, identični datom podskupu V′,
- E′ ⊆ E skup grana koje povezuju parove čvorova iz V′, a koje su prisutne u izvornom grafu G

Drugim riječima, podgraf indukovan skupom čvorova V′ sadrži **tačno one grane iz originalnog grafa** koje povezuju bilo koja dva čvora iz V′. Nema dodatnih grana niti čvorova.



$G' = G\,[\{v_1, v_2, v_4\}]$

Definicija izomorfizma

Dva grafa G1=(V1,E1) i G2=(V2,E2) su **izomorfni** ako postoji bijektivna funkcija f:V1→V2 takva da:

$$(u,v) \in E1 \Leftrightarrow (f(u),f(v)) \in E2$$

Drugim rečima, struktura grafa G1 se može preslikati na graf G2 bez promene relacija između čvorova.

Primer dva izomorfna grafa

Graf G1G_1:

- Skup čvorova: $V1=\{a,b,c,d\}$
- Skup grana: $E1=\{(a,b),(b,c),(c,d),(d,a),(a,c)\}$

Graf G2G_2:

- Skup čvorova: $V2=\{1,2,3,4\}$
- Skup grana: $E2=\{(1,2),(2,3),(3,4),(4,1),(1,3)\}$

Izomorfizam

Funkcija f koja definiše izomorfizam između G1 i G2 je:

$f(a)=1,f(b)=2,f(c)=3,f(d)=4$

Preslikavanje grana:

- $(a,b)\in E1 \rightarrow (f(a),f(b))=(1,2)\in E2$
- $(b,c)\in E1 \rightarrow (f(b),f(c))=(2,3)\in E2$
- $(c,d)\in E1 \rightarrow (f(c),f(d))=(3,4)\in E2$
- $(d,a)\in E1 \rightarrow (f(d),f(a))=(4,1)\in E2$
- $(a,c)\in E1 \rightarrow (f(a),f(c))=(1,3)\in E2$

Očigledno je da su G1 i G2 izomorfni.

Uslovi da grafovi budu izomorfni

1. **Jednak broj čvorova:** $|V1|=|V2|$
2. **Jednak broj grana:** $|E1|=|E2|$
3. **Stepen svakog čvora mora biti očuvan:** Ako čvor $u\in V1$ ima stepen k, tada $f(u)\in V2$ mora imati isti stepen k.
4. **Očuvanje susedstva:** Ako su čvorovi u i v susedi u G1, njihovi odgovarajući čvorovi f(u) i f(v) moraju biti susedi u G2.

Izomorfizam grafova znači da dva grafa imaju identičnu strukturu, ali različite oznake čvorova. Ovi uslovi su neophodni (a često i dovoljni) da se utvrdi izomorfizam.

Primera dva neizomorfna grafa koji ispunjavaju potrebne ove uslove:

Posmatrajmo grafove G1 = (V1,E1) i G2 = (V2,E2).

Graf G1:

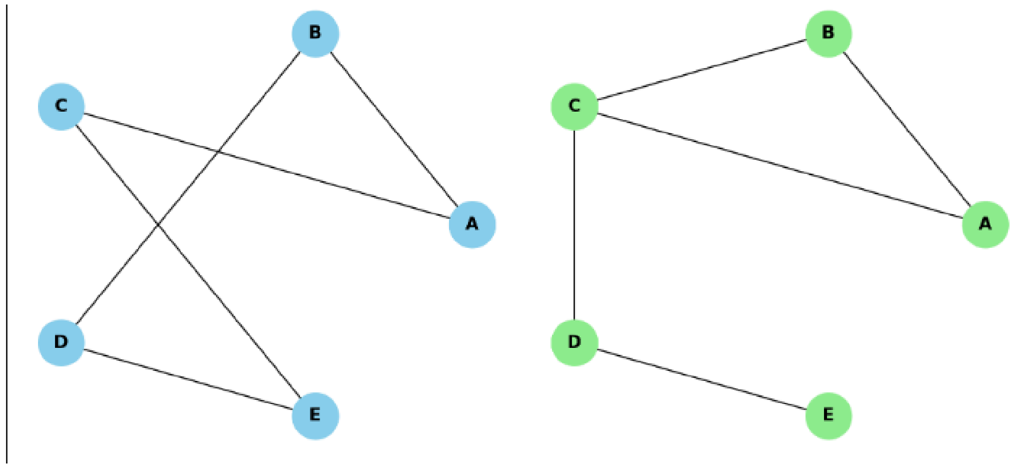
- Čvorovi: $V1=\{A,B,C,D,E\}$
- Grane: $E1=\{(A,B),(A,C),(B,D),(C,E),(D,E)\}$

Graf G2:

- Čvorovi: $V2=\{A,B,C,D,E\}$
- Grane: $E2=\{(A,B),(A,C),(B,C),(D,E),(C,D)\}$

Provera:

1. **Broj čvorova:** $|V1|=|V2|=5$
2. **Broj grana:** $|E1|=|E2|=5$
3. **Stepeni čvorova:**
   - U G1: Stepen svakog cvora je 2.
   - U G2: Stepen svakog cvora je 2.
4. **Različita struktura:**



Grafovi nisu izomorfni jer su im strukture različite. Nije ispunjeno očuvanje susedstva.

Relacija “je izomorfan” je relacija ekvivalencije nad skupom svih prostih grafova.

Da bismo dokazali da je relacija "je izomorfan sa" na skupu svih prostih grafova relacija ekvivalencije, moramo pokazati da zadovoljava tri osobine:

1. Refleksivnost
2. Simetričnost
3. Tranzitivnost

1. Refleksivnost

Relacija R na skupu S je refleksivna ako za svako $a \in S$ važi aRa.
U našem slučaju:
Svaki graf G je izomorfan sam sebi.
Naime, postoje preslikavanja $f:V(G)\to V(G)$ (gdje V(G) predstavlja skup čvorova grafa G) koja je bijekcija, a istovremeno čuva ivice.
Dakle, G je izomorfan sa G.

2. Simetričnost

Relacija R na skupu S je simetrična ako za svaka dva elementa $a,b \in S$ važi:
ako je a u relaciji R sa b, onda je b u relaciji R sa a.
U našem slučaju:
Ako je graf G1 izomorfan grafu G2, tada postoji bijekcija $f:V(G1)\to V(G2)$ koja čuva ivice.
Obrnuto preslikavanje $f^{-1}:V(G2)\to V(G1)$ takođe je bijekcija koja čuva ivice (jer je obrnuto preslikavanje bijekcije takođe bijekcija i čuva strukturu).
Dakle, ako je G1 izomorfan G2, tada je i G2 izomorfan G1

3. Tranzitivnost

Relacija R na skupu S je tranzitivna ako za svaka tri elementa $a,b,c \in S$ važi:
ako aRb i bRc, onda aRc.
U našem slučaju:
Ako je graf G1 izomorfan sa grafom G2 i graf G2 izomorfan sa grafom G3, tada:

- Postoji bijekcija $f:V(G1)\to V(G2)$ koja čuva ivice.
- Postoji bijekcija $g:V(G2)\to V(G3)$ koja čuva ivice.

Sada posmatramo kompoziciju funkcija $h=g \circ f : V(G1)\to V(G3)$

- h je bijekcija jer je kompozicija bijekcija takođe bijekcija.
- h čuva ivice jer su f i g funkcije koje čuvaju ivice.

Dakle, G1 je izomorfan G3.

Zaključak:

Relacija "je izomorfan sa" je relacija ekvivalencije.

.

Algoritam za proveru izomorfizma dva prosta grafa

Prosti grafovi $G1=(V1,E1)$ i $G2=(V2,E2)$ su **izomorfni** ako postoji bijektivno preslikavanje $f:V1\to V2$ tako da za svaka dva čvora $u,v\in V1$:

$\{u,v\}\in E1 \Leftrightarrow \{f(u),f(v)\}\in E2$

Koraci algoritma

1. **Proverite osnovne uslove**:
   - Broj čvorova mora biti isti: $|V1|=|V2|$.
   - Broj grana mora biti isti: $|E1|=|E2|$.
   - Ako se ovi uslovi ne zadovolje, grafovi nisu izomorfni.
2. **Uporedite stepen čvorova**:
   - Izračunajte stepen svakog čvora u oba grafa.
   - Sortirajte stepene za G1.
   - Ako sortirani nizovi stepena nisu isti, grafovi nisu izomorfni.
3. **Generišite sve permutacije čvorova**:
   - Ako prethodni koraci ne isključuju izomorfizam, generišite sve bijekcije (permutacije) čvorova iz V1 na V2.
4. **Proverite očuvanje grana za svaku permutaciju**:
   - Za svaku permutaciju $f:V1\to V2$, proverite da li za svaku granu $\{u,v\}\in E1$, važi $\{f(u),f(v)\}\in E2$.
   - Ako pronađete neku permutaciju za koju ovo važi, grafovi su izomorfni.
5. **Zaključak**:
   - Ako nijedna permutacija ne zadovoljava uslov iz koraka 4, grafovi nisu izomorfni.

Implementacija u pythonu:

```python
import itertools

def are_graphs_isomorphic(V1, E1, V2, E2):
    # 1. Proverite osnovne uslove
    if len(V1) != len(V2) or len(E1) != len(E2):
        return False

    # 2. Proverite stepen čvorova
    def degree_sequence(V, E):
        degrees = {v: 0 for v in V}
        for u, v in E:
            degrees[u] += 1
            degrees[v] += 1
        return sorted(degrees.values())

    if degree_sequence(V1, E1) != degree_sequence(V2, E2):
        return False

    # 3. Generišite sve permutacije
    for perm in itertools.permutations(V2):
        mapping = dict(zip(V1, perm))
        # 4. Proverite očuvanje grana
        permuted_edges = {(mapping[u], mapping[v]) for u, v in E1}
        if permuted_edges == set(E2):
            return True

    # 5. Grafovi nisu izomorfni
    return False
```