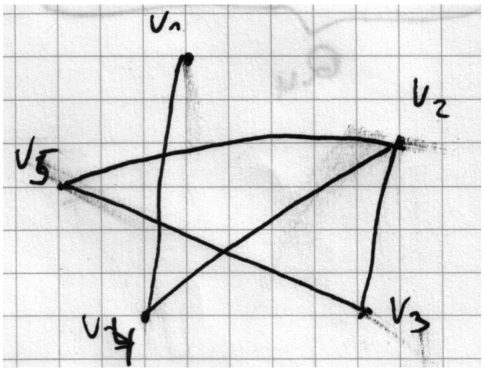
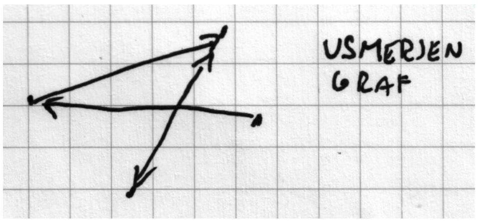
Teorija grafov

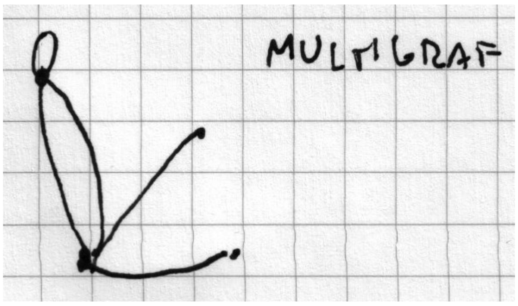
Definicija: **Graf** je urejen par $G=(V,E)$, kjer je V neprazna **množica točk** oziroma vozlišč, E pa množica povezav. Vsaka povezava je množica dveh različnih vozlišč.



$V=\{v_1, v_2, v_3, v_4, v_5\}$
$E=\{\{v_1, v_4,\}, \{v_3, v_5\}, \ldots\}=\{v_1 v_4, v_3 v_{5,} \ldots\}$
$v_1 v_4 \equiv v_4 v_3$

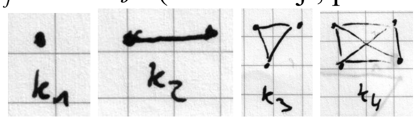
- **Usmerjeni grafi** imajo povezave lahko usmerjene.
- **Multigrafi** dovoljujejo vzporedne povezave (več povezav med dvema točkama).



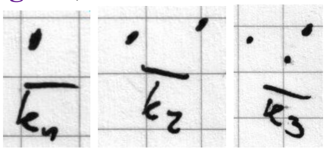


Nekatere standardne družine grafov

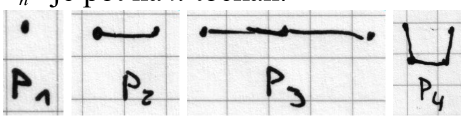
- **Polni grafi** (complete graphs); $K_{1,} K_{2,} \ldots$: vsaka točka je povezana z vsako
  $V(K_n)=\{v_1, v_2, v_3, \ldots v_n\}$
  $v_i \sim v_j$, ko $i \neq j$ (sta sosednji, povezani)



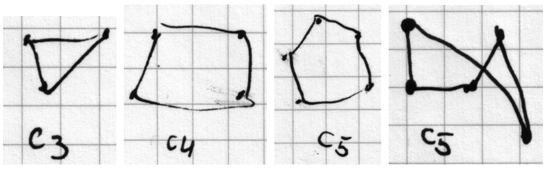
- **Prazni grafi**; $\overline{K_1}, \overline{K_2}$: ni nobenih povezav



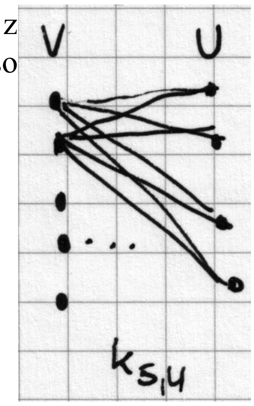
- **Poti**; P_n je pot na n točkah.



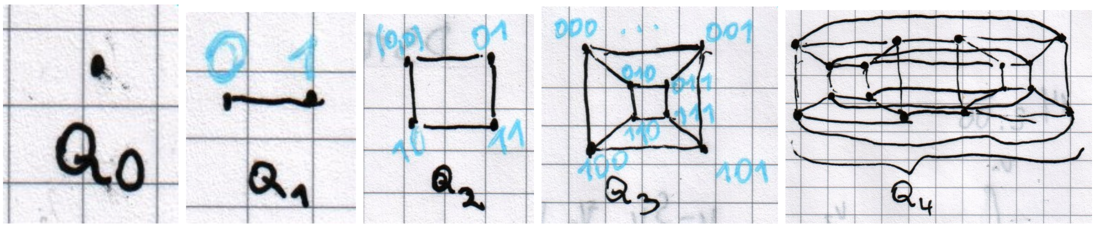
- **Cikli**; $C_n; n \geq 3$: so poti z začetkom in koncem v isti točki



- **Polni dvodelni grafi**; $K_{m,n}$: vsaka izmed prvih m točk je povezana z vsako od n točk na drugi strani, točke na isti strani pa med seboj niso povezane.



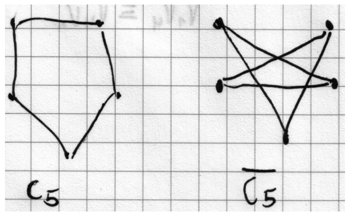
- **Hiperkocke**: vedno naredimo dve kopiji prejšnje kocke in naredimo povezave istoležnih točk. $V(Q_d)=\{0,1\}\times\{0,1\}\times\ldots\times\{0,1\}=\{0,1\}^d=\{(a_1,a_2,\ldots,a_d);a_i\in\{0,1\}\}$
  $u,v\in Q_d$. $u\sim v\Leftrightarrow$ se razlikujeta v natanko eni komponenti



Operacije z grafi

Komplement grafa

$\overline{G}$



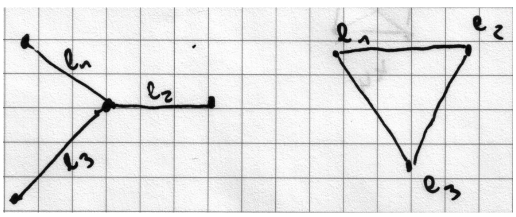
Ta dva grafa sta izomorfna!

Definicija: $V(\overline{G})=V(G)$. $u \underset{\overline{G}}{\sim} v \Leftrightarrow \neg(u \underset{G}{\sim} v)$

Če je G graf na n točkah, velja $G\cup\overline{G}=K_n$, $G\cap\overline{G}=\overline{K_n}$.

Povezavni graf

$L(G)$ (line graph)



Če imamo graf z množico povezav E, je v $L(G)$ to množica točk, povezave pa tvorimo med točkami, ki so kot povezave v prejšnjem grafu imele skupno krajišče.

Matematično: $V(L(G))=E(G)$.

$e \underset{L(G)}{\sim} f \Leftrightarrow e\neq f \wedge e\cap f=\emptyset$ (imata skupno krajišče)

Stopnja grafa

Definicija: **Stopnja točke** $d(x); v \in V(G)$ je število povezav, ki imajo v za krajišče.

Minimalna stopnja: $\delta(G)$. **Maksimalna stopnja**: $\Delta(G)$.

Definicija: Graf G je d-regularen, če velja: $\delta(G)=\Delta(G)=d$. *Vsa vozlišča so stopnje* d.

- $K_{m,n}$ je regularen; $m=d$
- Q_d je d-regularen
- C_n je 2-regularen

Matrike grafov

Imamo graf G: $V(G)=\{v_1, v_2, \ldots, v_n\}$. $E(G)=\{e_1, e_2, \ldots, e_n\}$

Matrika sosednosti *A*(*G*)

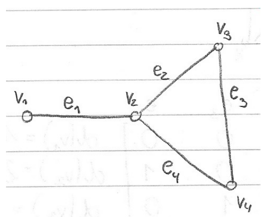
Je simetrična ($n \times n$).

$$A(G)=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \ldots & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & \ldots & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & \ldots & a_{3n} \\ . & . & . & . & \ldots & . \\ . & . & . & . & \ldots & . \\ . & . & . & . & \ldots & . \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & \ldots & a_{nn} \end{bmatrix} ; \quad a_{ij}=\begin{cases} 1 ; & v_i \sim v_j \\ 0 ; & \text{sicer} \end{cases}$$

$$A(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_n \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & \ldots & v_n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \ldots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \ldots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \ldots & a_{nn} \end{bmatrix}_{n \times n} \end{array}$$

S to matriko primerjamo sosednost vozlišč (na diagonali je vedno 0 – vozlišče ni sosedno samo s seboj).

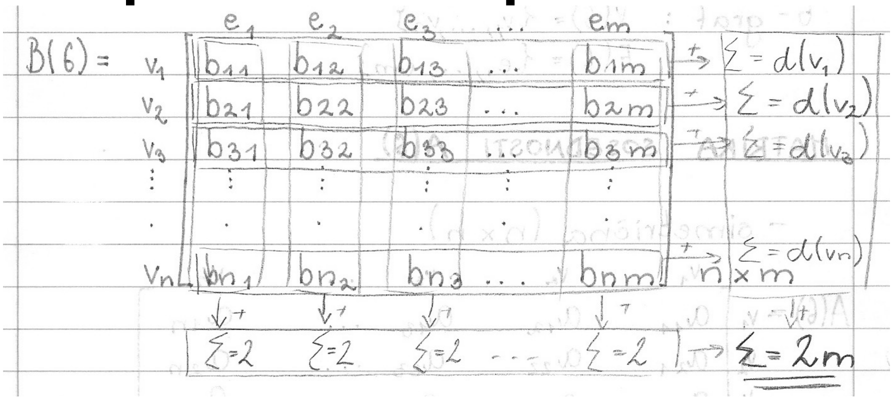
Primer:



$$A(G)=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

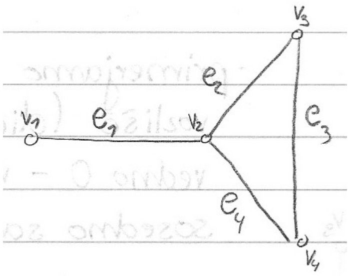
Incidenčna matrika *B*(*G*)

$$B(G)=\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & \dots & b_{1m} \\ b_{21} & b_{22} & b_{23} & b_{24} & \dots & b_{2m} \\ b_{31} & b_{32} & b_{33} & b_{34} & \dots & b_{3m} \\ . & . & . & . & \dots & . \\ . & . & . & . & \dots & . \\ . & . & . & . & \dots & . \\ b_{n1} & b_{n2} & b_{n3} & b_{n4} & \dots & b_{nm} \end{bmatrix} \quad ; \quad b_{ij}=\begin{cases} 1\,; v_i \in e_j \\ 0\,;\ \text{sicer} \end{cases}$$



V tej matriki vidimo, ali povezava vsebuje vozlišče ali ne.

Primer:



$$B(G)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \dots \quad \begin{matrix} d(v_1)=1 \\ d(v_2)=2 \\ d(v_3)=2 \\ d(v_4)=2 \end{matrix}$$

Zapis: $|v|$ - število točk , $|e|$ - število povezav

Trditev: $\sum_{v\in V} d(v)=2|e|$. Če je *G* *k*-regularen, potem $k\cdot|V|=2|e|$.

$$Q_d \;:\; n=2^d\,, d\text{-regularen} \;.\; d\cdot 2^d=2|e| \;.\; \frac{d\cdot 2^d}{2}=|e| \;.\; |e|=d\cdot 2^{d-1} \;.$$

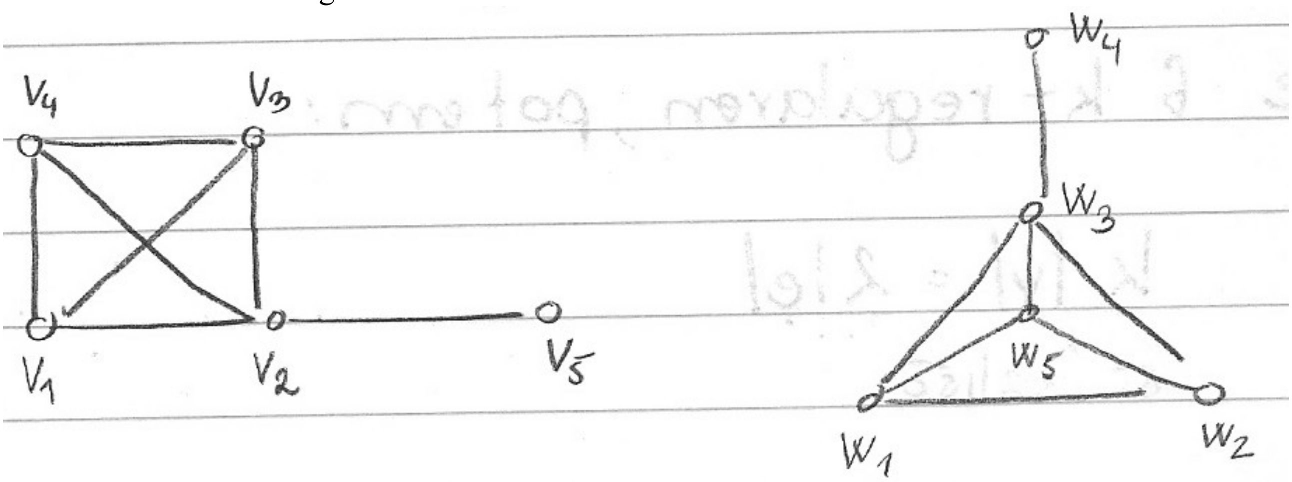
Trditev: *v* je soda točka, če je $d(v)$ sodo število in je liha točka, če je $d(v)$ liho število.

Trditev: Vsak graf ima sodo mnogo lihih točk.

Dokaz: $V=V_L\cup V_S$. $2|e|=\sum_{v\in V_S} d(v)+\sum_{v\in v_L} d(v)$. $2|e|-\sum_{v\in V_S} d(v)=\sum_{v\in v_L} d(v)$. Prva dva člena sta zagotovo soda, zato mora biti tudi tretji (desna stran enačbe). Če seštevamo liho število, jih moramo sešteti sodo mnogo, da dobimo sodo število.

Izomorfizem grafov

Primer dveh izomorfnih grafov:

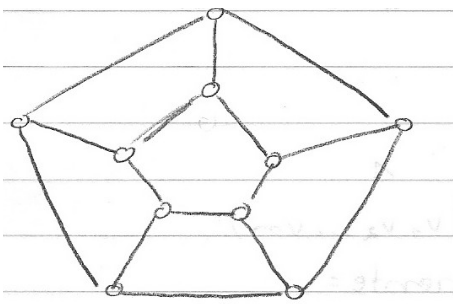
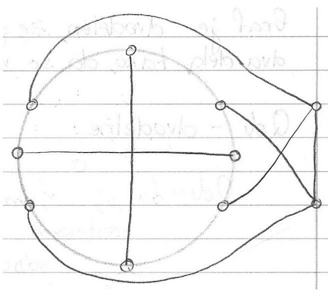


Definicija: Grafa G_1 in G_2 sta izomorfna, če obstaja preslikava $f: V(G_1) \to V(G_2)$ tako, da velja:

- f je bijekcija
- $u \underset{G_1}{\sim} v \Leftrightarrow f(u) \underset{G_2}{\sim} f(v)$

f za gornji primer bi bil $f(v_1)=w_1; f(v_2)=w_3; f(v_3)=w_2; f(v_4)=w_5; f(v_5)=w_4$.

Petersonov graf:


...


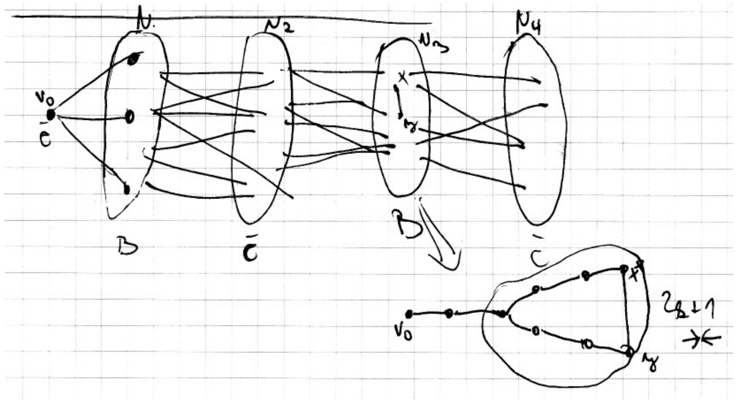
Definicija: Graf je **dvodelen**, če lahko točke pobarvamo z dvema barvama (črna in bela) tako, da ima vsaka povezava raznobarvni krajišči.

- C_n je dvodelen ntk. je n sodo število.
- $K_{n,m}$ je v vsakem primeru dvodelen.
- Q_d so dvodelne.
-

Trditev: Graf G je **dvodelen** natanko tedaj, ko G ne vsebuje lihega cikla.

Dokaz:

- ⇒ Če je G dvodelen, nima lihega cikla. Recimo, da je C nek lih cikel dolžine $2k+1$ v G. C ni 2-obarvljiv, zato G ni 2-obarljiv.
- ⇐ Če graf G nima lihega cikla, je dvodelen. Začnimo barvati graf s črno in belo barvo v neki točki. Če sta povezani dve točki iste barve, tvorita lih cikel (glej sliko na desni). Prišli smo v protislovje, saj graf G nima lihih ciklov.



Grafično zaporedje

Definicija: Končno zaporedje po velikosti urejenih naravnih števil $d_1 \leq d_2 \leq d_3 \leq \ldots \leq d_n$ (zaporedje je $d_1, d_2, d_3, \ldots, d_n$) je **grafično**, če obstaja graf z n vozlišči, ki imajo stopnje enake številom v tem zaporedju ($d_1, d_2, d_3, \ldots, d_n$).

Zgled: Zaporedje 0, 1, 2, 2, 3 je grafično.

Uporabimo „požrešno metodo“:

- Vozlišče z največjo stopnjo d_n povežemo z naslednjimi d_n vozlišči po vrsti. Tem vozliščem stopnje zmanjšamo za 1. Zadnje vozlišče izpustimo.
- Stopnje ponovno uredimo po velikosti.
- Postopek ponovimo s tako dobljenim zaporedjem.

Postopek se lahko konča na več različnih načinov:

- Dobimo same 0. Zaporedje <u>je</u> grafično.
- Zmanjka vozlišč. Zaporedje <u>ni</u> grafično.
- Stopnja nekega vozlišča se zmanjša pod 0. Zaporedje <u>ni</u> grafično.

Primera:

- 1, 2, 3, 4, 4, 4 ... 1, 1, 2, 3, 3 ... 0, 1, 1, 2 ... 0, 0, 0. Zaporedje je grafično. Če vozlišča oštevilčimo, lahko sproti tudi rišemo graf.
- 1, 2, 4, 4, 4, 5 ... 0, 1, 3, 3, 3 ... 0, 0, 2, 2 ... 0, -1, 1. Zaporedje ni grafično.

Izrek (ki omogoča / dokazuje omenjeno požrešno metodo): Zaporedje $d_1 \leq d_2 \leq d_3 \leq \ldots \leq d_n$ je grafično natanko takrat, ko je zaporedje $d_1, d_2, \ldots, d_{n-d_n-1}, d_{n-d_n}-1, d_{n-d_n+1}-1, \ldots, d_{n-1}-1$

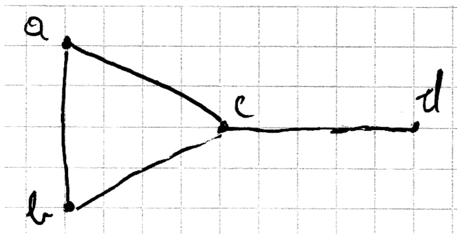
grafično.

Podgrafi

Definicija: H in G sta grafa. H je **podgraf** grafa G, če velja $V(H) \subseteq V(G)$ in $E(H) \subseteq E(G)$. Zapišemo $H \subseteq G$.

- H je **vpet podgraf** grafa G, če $H \subseteq G \wedge V(G) = V(G)$.
- H je **induciran podgraf** grafa G, če $H \subseteq G \wedge \forall e = xy : (x, y \in V(H) \Rightarrow e \in E(H))$ (induciran po točkah). Induciran podgraf je natanko določen za vsako izbrano množico točk. Zapis $H = G[U]$ pomeni, da je H induciran podgraf grafa G, U pa je množica točk grafa H.

Primer: *Koliko podgrafov ima graf na sliki (desno)?* Vzemimo grafe s posameznim številom točk:



- $|V(H)| = 1$: **4**
- $|V(H)| = 2$: $\binom{4}{2} = 6$ možnih parov točk. Nekatere so lahko povezane (*ab*, *ac*, *bc*, *cd*), zato jih štejemo dvakrat, nekatere pa ne (*ad, bd*). Možnosti je **10**.
- $|V(H)| = 3$: $\binom{4}{3} = 4$ možnih parov točk. Možnosti je **18**.
- $|V(H)| = 4$: Vzamemo cel graf, možnost je **16**.

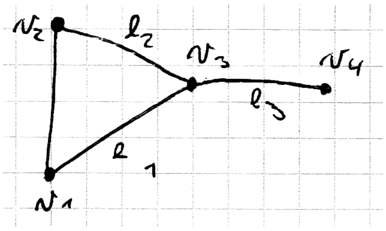
Izbran graf ima **<u>48</u>** podgrafov.

Sprehodi po grafu

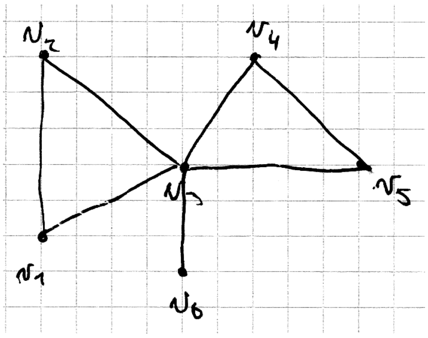
Definicija: Zaporedje točk in povezav $v_0 e_1 v_1 e_2 v_2 \ldots\ldots v_{k-1} e_k v_k$, kjer je $e_i = v_{i-1} v_i$, imenujemo **sprehod**.

- **Dolžina sprehoda** je enaka številu povezav, preko katerih gremo.
- Sprehod je **enostaven**, če so vse uporabljene povezave različne (vsako povezavo uporabimo kvečjemu enkrat).
- Enostaven sprehod je **pot**, če so vse obiskane točke različne (nikoli ne pridemo v točko, ki smo jo že obiskali).
- **Obhod** je sklenjen sprehod (tak, ki se konča v začetni točki).
- **Cikel** je sklenjena pot (obhod, pri katerem dvakrat obiščemo le začetno/končno točko).

Zgleda:

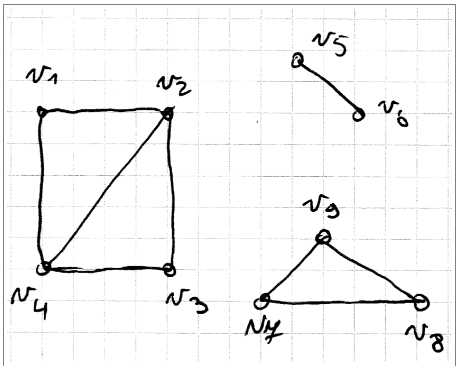
- 

  Sprehod v tem grafu bi bil lahko $v_1 e_1 v_2 e_2 v_3 e_2 v_3 e_2 v_3 e_3 v_4$. Ker je graf enostaven, nam ni treba navajati povezav: $v_1 v_2 v_3 v_2 v_3 v_4$. Dolžina tega sprehoda je 5. To ni enostaven sprehod. $v_1 v_2 v_3 v_2 v_3 v_4 v_3 v_1$ je v tem grafu obhod.

- 

  V tem grafu je $v_1 v_2 v_3 v_4 v_5 v_3 v_6$ je enostaven sprehod, vendar ni pot.

Trditev: **Če** v grafu *G* **obstaja sprehod** od *u* do *v*, **potem obstaja pot** od *u* do *v*.
Zakaj to drži? Če velja $v_i = v_j$, lahko celotno vmesno zanko izbrišemo:
$v_1 v_2 v_3 \dots v_i \cancel{v_{i+1} v_{i+2} \dots v_{j-1} v_j} v_{j+1} v_{j+2} \dots$.



Primer nepovezanega grafa.

Definicija: Graf *G* je **povezan**, če za poljubni vozlišči $u, v \in V(G)$ obstaja sprehod (oziroma pot) od *u* do *v*.

Definicija: Definiramo relacijo P nad $V(G)$: $uPv \Leftrightarrow \exists$ sprehod $u \rightarrow v$. Taka relacija graf razbije na (povezane) komponente.

Primer v grafu na desni:
$V_1 = \{v_1, v_2, v_3, v_4\}, V_2 = \{v_5, v_6\}, V_3 = \{v_7, v_8, v_9\}$.
Komponente tega grafa so $G[V_1], G[V_2], G[V_3]$.

Definicija: Naj bo *G* povezan graf in $u, v \in V(G)$. $d(u, v)$ je **dolžina najkrajše poti** med *u* in *v* (t.j. **razdalja** med *u* in *v*).

Lastnosti:

1. $\forall x : d(x, x) = 0$
2. $\forall x, y : d(x, y) = d(y, x)$

3. $\forall x, y, z : d(x, y) + d(y, z) \geq d(x, z)$ (trikotniška neenakost)

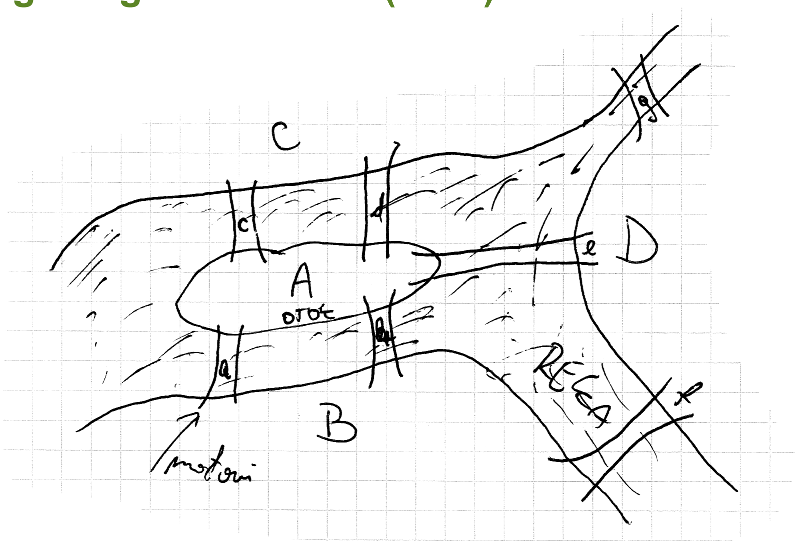
Zaradi teh treh lastnosti pravimo, da je funkcija d **metični prostor**.

Definicija: Naj bo G povezan graf. **Premer grafa** G je $diam(G) = \max_{u, v \in V(G)} d(u, v)$.

Lastnosti:

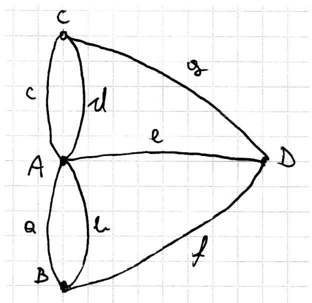
- $diam(K_n) = 1$
- $diam(P_n) = n - 1$
- $diam(C_n) = \left\lfloor \frac{n}{2} \right\rfloor$
- $diam(Q_d) = d$: Če imamo točki $u = (u_1, u_2, \ldots, u_d)$, $v = (v_1, v_2, \ldots, v_d)$, je razdalja enaka številu komponent, v katerih se točki razlikujeta.

Problem Königsbergovih mostov (1730)



Problem: Ali obstaja obhod mesta, ki bi prečkal vsak most natanko enkrat in se vrnil v izhodišče?

V teoriji grafov bi problem opisali tako: *ali obstaja enostaven obhod v grafu, ki uporabi vse povezave?* Odgovor je ne, saj bi morala vsaka točka imeti sodo stopnjo.



Definicija: Sprehod/Obhod v grafu je **Eulerjev**, če vsebuje vsako povezavo natanko enkrat.

Eulerjev ***izrek***:

- Graf *G* **ima Eulerjev obhod** natanko takrat, ko je *G* povezan in je vsaka točka sode stopnje.
- Graf *G* **ima Eulerjev sprehod** natanko takrat, ko je *G* povezan in ima največ dve točki lihe stopnje.

Povedano drugače:
Izrek (*): Naj bo G povezan graf.

- G ima Eulerjev obhod natanko takrat, ko so vse točke sode stopnje.
- G ima Eulerjev sprehod natanko takrat, ko ima največ dve točki lihe stopnje.

Definicija: **Most** je povezava, ob brisanju katere graf razpade na dva nepovezana dela.

Fleurjev algoritem

Poišče Eulerjev obhod v G s samimi točkami sode stopnje.

1. Izberemo začetno točko.
2. Prečkaj poljubno povezavo, le most izberi samo, kadar ni na voljo nobene druge povezave.
3. Prehojeno povezavo odstrani, tudi točke, ki so ostale brez povezav.
4. Končaj, ko ni več nobene povezave.

Algoritem deluje (saj zaradi neizbiranja mostov ne moremo dobiti dveh nepovezanih delov grafa). To ***dokazuje*** resničnost prvega dela gornjega izreka (*).

Kako dokažemo drug del izreka? Imamo dve lihi točki. Če med njima dodamo namišljeno povezavo, so vse točke sode. Z gornjim algoritmom lahko najdemo Eulerjev obhod. Če eno povezavo (torej prej dodano) zbrišemo, nastane sprehod (vendar še vedno obišče vse točke). Graf ima Eulerjev sprehod.

Zgled: (slika 8) Nariši graf z eno potezo. *Graf lahko narišemo z eno potezo, če obstaja Eulerjev sprehod!*

Izrek: Število potez, ki jih potrebujemo, da narišemo (povezan) graf je enako $\frac{\text{število lihih točk}}{2}$ oziroma ena poteza, če so vse točke sode.

Zgled: Nariši graf (slika 9). Potrebujemo dve potezi, saj so 4 točke lihe stopnje.

Fleurjev algoritem *na poljubnem povezanem grafu* za risanje grafa s čim manj potezami uporabimo tako, da umetno dodamo povezave med pari točk z lihimi stopnjami.

Če dobljen obhod predstavimo v ciklu (kjer se točke ponavljajo) in brišemo prej dodane povezave, graf razpade na več odsekov. Ti predstavljajo poteze.

Problem kitajskega poštarja (PKP)

Iščemo najkrajši obhod poljubnega povezanega grafa, ki prečka vsako povezavo vsaj enkrat.

Trditev: Poštar nikoli ne gre skozi povezavo več kot dvakrat.

Dokaz: (slika 10)
Prvotna pot: $p_1 e\, p_2 e\, p_3 e\, p_4$
Optimiziramo pot: $p_1\, p_3 e\, p_2\, p_4$

$E = E_1 \cup E_2$
$PKP = |E_1| + 2|E_2| = |E| + |E_2|$ (pomeni, da imam čim manj povezav z 2 prehodi)
$G\big(V(G), E(G) \setminus E_2\big)$ je največji vpet Eulerjev v *G*.

Najdi PKP za (slika 11)!
Način: Podvojimo poti med pari lihih točk tako, da vsako liho točko porabimo enkrat in je vsota dolžin poti najkrajša. Dobimo graf s samimi točkami sode stopnje in uporabimo Fleurjev algoritem.

Hamiltonovi cikli/poti

Definicija: *Vpetemu ciklu v grafu* pravimo **Hamiltonov cikel**.

Iščemo torej cikel, ki obišče vse točke (a vsako samo enkrat).

Primer za hiperkocko: (slika 12)

Definicija: *Vpeti poti* v grafu pravimo **Hamiltonova pot**.

Iščemo torej pot, ki obišče vse točke (a vsako samo enkrat).

Graf $K_{2,3}$ (slika 13) ima Hamiltonovo pot, nima pa Hamiltonovega cikla.

Tudi ta graf ima samo Hamiltonovo pot (slika 14).

Kaj pa slika 15? Izkaže se, da ima Hamiltonov cikel.

V splošnem imajo $K_{m,n}$ Hamiltonovo pot, če $|m-n| \leq 1$ in cikel, če $m=n$. Nepolni dvodelni grafi, če imajo Hamiltonov cikel/pot, imajo enake lastnosti.

Zgodovina problema: Sir William Hamilton (1805 – 1865) je bil vodilni matematik svojega časa. Izumil je igro „*potovanje okoli sveta*“ (glej gornjo sliko). Problem: poiskati Hamiltonove cikle z danimi začetnimi 5 črkami, npr. BCPNM (Hamilton je dokazal, da je to možno za poljubno začetno zaporedje).

Primer: Problem skakačevega sprehoda

Slika 16.

Skakač želi obiskati vsa polja, vsako samo enkrat, in to v ciklu.

To je problem Hamiltonovega cikla, če polja na šahovnici prevedemo v točke, možne skoke pa v povezave.

Primer: Gryeve kode so opisi Hamiltonovih ciklov v hiperkockah.

000 d=3
001
011
010
110
111
101
100
000

Primer: Problem trgovskega potnika: Trgovski potnik želi obiskati nekaj mest (obtežen poln graf) in se vrniti v začetno mesto, tako da bo vsako mesto obiskal natanko enkrat in je skupni strošek potovanja najmanjši.

Gre za zelo praktičen problem, v katerega lahko prevedemo problem Hamiltonovih ciklov: neobstoječe povezave označimo z neskončno težo, ostale pa z 1.

Izrek (Ore, 1960): Naj bo *G* enostaven graf z vsaj tremi točkami. Če za poljubni nesosedni točki *u*, *v* iz *G* velja $d(u)+d(v) \geq \frac{|V(G)|}{2}+\frac{|V(G)|}{2}=|V(G)|$, potem je *G* Hamiltonov.

Izrek (Dirac): Če ima *G* vsaj tri točke in velja $\delta(G) \geq \frac{|V(G)|}{2}$, potem je *G* Hamiltonov.

Osnovni potrebni pogoj

Če iz povezanega grafa *G* odstranimo *k* točk in tako dobimo graf z več kot *k* komponentami, potem graf ne vsebuje Hamiltonovega cikla.

Če je komponent več kot $k+1$, potem v grafu *G* ni niti Hamiltonove poti.

Posledica: Naj bo *G* dvodelen povezan graf z dvodelnim razbitjem $V(G)=X \cup Y$. Če $|X| \neq |Y|$, potem *G* nima Hamiltonovega cikla. Če je $\left| |X|-|Y| \right| > 1$, potem G nima niti Hamiltonove poti.