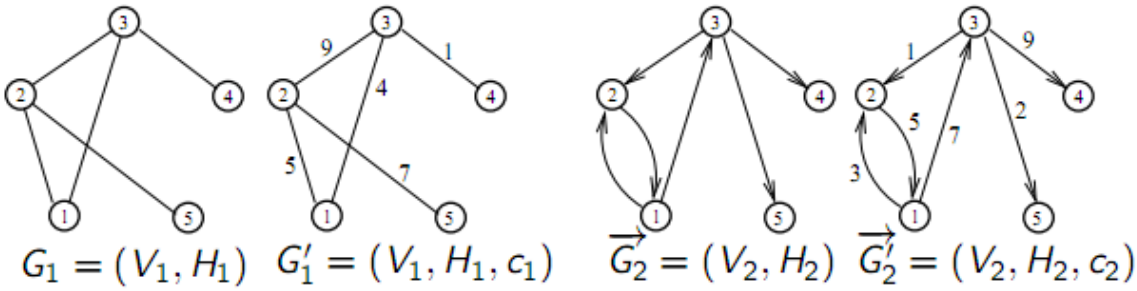
1. Základné pojmy teórie grafov. Graf, digraf, ďalšie štruktúry TG, podgraf, úplný graf, stupeň vrchola, počet vrcholov nepárneho stupňa, izomorfizmus, komplementárnosť, reprezentácia grafov.

- **Graf**om nazveme usporiadanú dvojicu G = (V, H), kde V je neprázdna konečná množina a H je množina neusporiadaných dvojíc typu {u, v} takých, že u ∈ V , v ∈ V a u 6= v, t. j.
- **Digraf**om nazveme usporiadanú dvojicu −→G = (V, H), kde V je neprázdna konečná množina a H je množina usporiadaných dvojíc typu (u, v) takých, že u ∈ V , v ∈ V a u 6= v, t. j.
  Prvky množiny V nazývame vrcholmi a prvky množiny H hranami grafu G / orientovanými hranami digrafu −→G .
- **Diagram** - grafická reprezentácia grafu (napr. rovinný diagram)
- Hovoríme, že graf G′ = (V′, H′) je **podgrafom** grafu G =(V, H), ak platí V′ ⊆ V a H′ ⊆ H. V tomto prípade budeme písať G′ ⊆ G. Digraf −→G′ = (V′, H′) je podgrafom digrafu −→G = (V,H), ak V′ ⊆ V a H′ ⊆ H.
- Hovoríme, že graf G′ = (V′, H′) je **faktorovým podgrafom** grafu G = (V, H), ak platí V′ = V a H′ ⊆ H. Analogicky definujeme faktorový podgraf digrafu −→G.
- Graf G = (V, H) nazveme **úplný**m **graf**om, ak množina H obsahuje všetky možné dvojice typu {u, v}, kde u, v ∈ V a u 6= v. Úplný graf o n vrcholoch budeme značiť Kn.
- **Stupeň** deg(v) **vrchola** v v grafe G = (V, H) je počet hrán incidentných s vrcholom v.
  V digrafe odeg(v) / ideg(v) - pocet hran vychadzajucich / vchadzajucich do v.
- Počet vrcholov nepárneho stupňa v ľubovoľnom grafe G = (V, H) je párny.
- Graf G = (V, H) je **izomorfný s grafom** G′ = (V′, H′), ak existuje také vzájomne jednoznačné zobrazenie f : V ↔ V′, že pre každú dvojicu vrcholov u, v ∈ V platí: {u, v} ∈ H práve vtedy, keď {f(u), f(v)} ∈ H′ .Zobrazenie f sa volá izomorfizmus grafov G a G′
- **Pravidelný graf stupňa k** je taký graf G = (V, H), v ktorom má každý vrchol v ∈ V stupeň k.
- Grafy G = (V, H), G = (V , H) nazveme **komplementárne**, ak V = V a pre každú dvojicu vrcholov u, v ∈ V takých, že u 6= v, platí: {u, v} ∈ H práve vtedy, keď {u, v} ∈/ H.

Reprezentácia grafov a digrafov

- **Reprezentácia diagramom grafu**



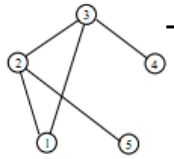
- **Reprezentácia množinami vrcholov a hrán**
  Nexh V_1 = {1,2,3,4,5}, H_1 = { {1,2}, {1,3}, {2,3}. {2,5}, {3,4}}.
  Množinami V_1 a H_1 je jednoznačne určený graf G_1 = (V_1 ,H_1).

  Podobne nech V_2 = {1,2,3,4,5} a
  H_2 = { (1,2), (1,3), (2,1), (3,2), (3,4), (3,5) },
  Potom množinami V_2, H_2 je jednoznačne určený digraf G_2 = (V_2, H_2).

- **Maticou priľahlosti**

Matica priľahlosti $\mathbf{M} = (m_{ij})$ je štvorcová matica typu $n \times n$, kde $n = |V|$ je počet vrcholov grafu, resp. digrafu G, ktorej prvky sú definované nasledovne:

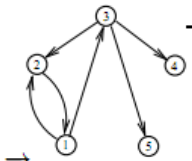
$$m_{ij} = \begin{cases} 1 & \text{ak } \{i,j\} \in H \\ 0 & \text{inak} \end{cases} \qquad m_{ij} = \begin{cases} 1 & \text{ak } (i,j) \in H \\ 0 & \text{inak} \end{cases} \tag{8}$$



$G_1 = (V_1, H_1)$

	1	2	3	4	5
1	-	1	1	-	-
2	1	-	1	-	1
3	1	1	-	1	-
4	-	-	1	-	-
5	-	1	-	-	-

Matica priľahlosti grafu G_1.



$\overrightarrow{G_2} = (V_2, H_2)$

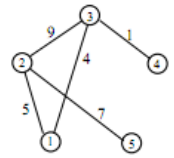
	1	2	3	4	5
1	-	1	1	-	-
2	1	-	-	-	-
3	-	1	-	1	1
4	-	-	-	-	-
5	-	-	-	-	-

Matica priľahlosti digrafu $\overrightarrow{G}_2$.

- **Maticou ohodnotení hrán**

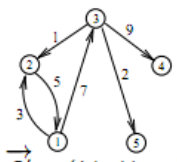
  Matica $\mathbf{M}$ ohodnotení hrán grafu, resp. digrafu je štvorcová matica typu $n \times n$, kde $n = |V|$ je počet vrcholov grafu, resp. digrafu a prvky ktorej sú definované nasledovne:

$$m_{ij} = \begin{cases} c(\{i,j\}) & \text{ak } \{i,j\} \in H \\ \infty & \text{inak} \end{cases} \qquad m_{ij} = \begin{cases} c((i,j)) & \text{ak } (i,j) \in H \\ \infty & \text{inak} \end{cases} \tag{9}$$



$G_1' = (V_1, H_1, c_1)$

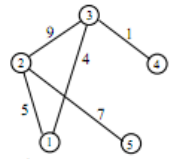
	1	2	3	4	5
1	-	5	4	-	-
2	5	-	9	-	7
3	4	9	-	1	-
4	-	-	1	-	-
5	-	7	-	-	-



$\overrightarrow{G_2'} = (V_2, H_2, c_2)$

	1	2	3	4	5
1	-	3	7	-	-
2	5	-	-	-	-
3	-	1	-	9	2
4	-	-	-	-	-
5	-	-	-	-	-

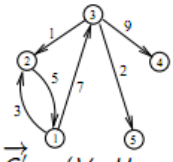
- **Zoznamom vrcholov okolia každého vrchola**



$G_1' = (V_1, H_1, c_1)$

$V(1)$	2	3	-
$V(2)$	1	3	5
$V(3)$	1	2	4
$V(4)$	3	-	-
$V(5)$	2	-	-

Vrcholy okolí pre graf G_1'.



$\overrightarrow{G_2'} = (V_2, H_2, c_2)$

$V^+(1)$	2	3	-
$V^+(2)$	1	-	-
$V^+(3)$	2	4	5
$V^+(4)$	-	-	-
$V^+(5)$	-	-	-

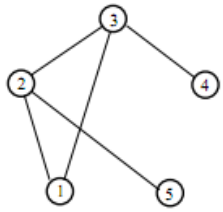
Vrcholy výstupných hviezd pre digraf $\overrightarrow{G_2'}$.

- **Incidenčnou maticou vrcholov a hrán**

  Incidenčná matica vrcholov a hrán je matica $\mathbf{B}$ typu $n \times m$, kde n je počet vrcholov a m počet hrán reprezentovaného grafu alebo digrafu. Každý prvok b_{ij} matice $\mathbf{B}$ hovorí o spôsobe incidencie vrchola i s hranou j nasledovne:

$$b_{ij} = \begin{cases} 1 & \text{ak vrchol } i \text{ je incidentný s hranou } j \text{ v grafe } G \\ 0 & \text{inak} \end{cases}$$

$$b_{ij} = \begin{cases} 1 & \text{ak vrchol } i \text{ je začiatočným vrcholom hrany } j \text{ v digrafe } \overrightarrow{G} \\ -1 & \text{ak vrchol } i \text{ je koncovým vrcholom hrany } j \text{ v digrafe } \overrightarrow{G} \\ 0 & \text{inak} \end{cases}$$



$G_1 = (V_1, H_1)$

v	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{2,5\}$	$\{3,4\}$
1	1	1			
2	1		1	1	
3		1	1		1
4					1
5				1	

Tabuľka: Incidenčná matica grafu $G_1 = (V_1, H_1)$

2. Cesty v grafoch. Sled, ťah, cesta, cyklus, polosled, poloťah, polocesta, polocyklus a ich analógie v digrafoch. Súvislosť grafov, Komponent grafu. Mosty a artikulácie. Tarryuho prieskum grafov.

- **Sled** v grafe G je ľubovoľná alternujúca (striedavá) postupnosť vrcholov a hrán tvaru μ(v1, vk) = (v1,{v1, v2}, v2,{v2, v3}, v3, . . . ,{vk−1, vk}, vk).
- **Ťah** v grafe G je taký sled v grafe G, v ktorom sa žiadna hrana neopakuje.
- **Cesta** v grafe G je taký sled v grafe G, v ktorom sa žiaden vrchol neopakuje. Pripúšťame aj tzv. triviálny sled, pre k = 1, t. j. sled tvaru (v1).
- **Cyklus** (orientovaný cyklus, polocyklus) je uzavretý ťah (orientovaný ťah, poloťah), v ktorom sa okrem prvého a posledného vrchola žiaden vrchol nevyskytuje viac než raz.
- **Polosled**, poloťah, polocesta (digraf).
- Hovoríme, že graf G = (V, H) je **súvislý**, ak pre každú dvojicuvrcholov u, v ∈ V existuje u–v cesta. Inak hovoríme, že graf G je nesúvislý.
- **Komponent grafu** G = (V, H) je jeho ľubovoľný maximálny súvislý podgraf.
- **Mostom** v grafe G = (V, H) nazveme takú hranu grafu G, po vylúčení ktorej vzrastie počet komponentov.
- **Artikuláciou** v grafe G nazveme taký vrchol, po vylúčení ktorého spolu s incidentnými hranami vzrastie počet komponentov.
- **Kružnica** je pravidelný súvislý graf 2. stupňa. Kružnicu o n vrcholoch budeme označovať Cn.
- **Tarryho algoritmus** na konštrukciu takého sledu v grafe G = (V, H), ktorý začína v ľubovoľnom vrchole s ∈ V , prejde všetkými hranami komponentu grafu G a skončí vo vrchole s. Výsledný sled budeme volať Tarryho sled.
    - Krok 1. Začni z ľubovoľného vrchola s ∈ V , polož u := s, T = (u). {T je inicializačne triviálny sled, obsahujúci jediný vrchol.}
    - Krok2. Ak môžeš, vyber k poslednému vrcholu u sledu T ďalšiu incidentnú hranu {u, v} podľa nižšie uvedených pravidiel T1, T2 a zaraď ju do sledu T . Zaznač si smer použitia hrany {u, v}. Ak doteraz vrchol v ešte nebol zaradený do sledu T , označ hranu {u, v} ako hranu prvého príchodu. Pri výbere hrany dodržuj nasledujúce pravidlá:
      T1: Každú hranu možno v jednom smere použiť iba raz
    - T2: Hranu prvého príchodu možno použiť, iba ak niet inej možnosti
      Krok 3. Ak taká hrana neexistuje – STOP. Inak polož u := v a pokračuj Krokom 2.

3. Najkratšia cesta. Základný algoritmus, Dijkstrov a Floydov algoritmus. Label set a label correct implementácie algoritmov najkratšej cesty.

- **Sled** (v1-vk sled) v grafe G je ľubovoľná alternujúca postupnosť vrcholov a hrán tvaru: mikro(v1, vk) = (v1, {v1,v2}, v2, {v2,v3}, v3 .... {vk-1,vk}, vk)
- **Ťah** v grafe G je taký v1-vk sled,v ktorom sa źiadna hrana neopakuje
- **Cesta** v grafe G je taký v1-vk sled, v ktorom sa žiaden vrchol neopakuje
- **triviálny sled** - sled tvaru (v1)
- **Orientovaný sled, ťah, cesta** - to isté ako hore len sú tam jednoduché zátvorky miesto grafu je použitý digraf

- **Polosled, tvar**: mikro(v1,vk) = (v1,h1,v2,h2,....vk-1,hk-1,vk), v ktorej je každá hrana hi incidentná s obomi susednmi vrcholmi tak , že jeden je začiatočný a druhý je koncovým vrcholom hrany h
- **Poloťah, polocesta** - v digrafe taký polosled ...
- **uzavretý sled** - ak $v_1=v_k$, inak nazveme otvorený
- **súvislý graf** - ak pre každú dvojicu vrcholov u,v existuje u-v cesta. Inak nesúvislý.
- **Komponent grafu** - ľubovoľný max podgraf
- **Most** - taká hrana po ktorej vylúčení sa zvýši počet komponentov
- **Artikulácia** - taký vrchol v grafe po ktorého vylúčení spolu s incidentnými hranami vzrastie počet komponentov
- **neorientovane súvislý digraf** - ak pre každú dvojicu vrcholov existuje v G u-v polosled. Inak je nesúvislý.
- **orientovane súvislý digraf** - ak pre každú dvojicu vrcholov u,v existuje aj orientovaný u-v sled alebo orientovaný v-u sled.
- **Základný algoritmus** na hľadanie najkratších orientoaných ciest:
    - Krok 1. Inicializácia.
    - Pre každý vrchol i ∈ V priraď dve značky t(i) a x(i*).{Značka t(i) predstavuje horný odhad dĺžky doteraz nájdenej najlepšej u–i cesty a x(i) jej predposledný vrchol.}*
    - Polož t(u) := 0, t(i) := ∞ pre i ∈ V, i != u a x(i) := 0 pre každé i ∈V
    - Krok 2. Zisti, či existuje orientovaná hrana (i , j) ∈ H, pre ktorú platí t(j) > t(i) + c(i,j)
    - Ak taká hrana (i , j) ∈ H existuje, potom polož t(j) := t(i) + c(i , j), x(j) := i a opakuj Krok 2.
    - Krok 3. Ak taká orientovaná hrana (z kroku 2.) neexistuje, STOP.
- **Dijkstrov algoritmus:**
    - Krok 1. Inicializácia. Pre každý vrchol i ∈ V priraď dve značky t(i) a x(i). Značky t(i) budú dvojakého druhu, a to dočasné (ktoré sa ešte v priebehu výpočtu môžu zmeniť) a definitívne (ktoré sa už nemôžu zmeniť)
        - Polož t(u) = 0, t(i) = ∞ pre i ∈ V, i != u a x(i) = 0 pre každé i ∈ V. Zvoľ riadiaci vrchol r := u a značku t( ) pri vrchole r = u prehlás za definitívnu, ostatné značky za dočasné.
    - Krok 2. Ak je r = v, STOP. Ak t(v) < ∞, značka t(v) predstavuje dĺžku najkratšej u–v cesty, ktorú zostroj spätne z v pomocou smerníkov x(i). Inak pre všetky hrany tvaru (r , j) ∈ H, kde j je vrchol s dočasnou značkou, urob: Ak t(j) > t(r) + c(r , j), potom t(j) := t(r) + c(r , j), x(j) := r Ponechaj zmenené značky ako dočasné.
    - Krok 3. Zo všetkých dočasne označených vrcholov nájdi ten vrchol i , ktorý má značku t(i) minimálnu. Značku pri tomto vrchole i prehlás za definitívnu a zvoľ za nový riadiaci vrchol r := i .
    - {Pokiaľ existuje viac vrcholov s rovnakou minimálnou značkou, akú má vrchol i, za definitívnu značku môžeme prehlásiť len značku pri jednom z týchto vrcholov – ten potom berieme za riadiaci. Na tie ďalšie dôjde v nasledujúcich krokoch výpočtu.} GOTO Krok 2.
- Algoritmus 3.6. **Label-set a Label-correct** implementácia algoritmu na hľadanie najkratších orientovaných u–v ciest z pevného vrchola u ∈ V do všetkých ostatných vrcholov v ∈ V v hranovo ohodnotenom digrafe $\vec{G}$ =(V, H, c) s nezápornou cenou orientovanej hrany c(h).
    - Krok 1: Inicializácia. Polož t(u) := 0, t(i) := ∞ pre i ∈ V, i 6= u a x(i) := 0 pre každé i ∈ V . Polož E := {u}.
    - Krok 2: Vyber r ∈ E, polož E := E − {r}. Pre všetky orientované hrany tvaru (r, j) ∈ H urob: Ak t(j) > t(r)+c(r, j), potom t(j) := t(r)+c(r, j), x(j) := r, E := E ∪{j}.
    - Krok 3: Ak E 6= ∅, choď na Krok 2. Ak E = ∅, potom t(i) predstavuje dĺžku najkratšej orientovanej u–i cesty pre každý vrchol i. Najkratšiu orientovanú u–i cestu zostroj potom spätne pomocou značiek x(i) ako v predchádzajúcich dvoch algoritmoch

4. Stromy a ich vlastnosti. Veta o ekvivalentných výrokoch s výrokom \\\"graf G je stromom\\\". Prehľadávanie grafu do šírky a do hĺbky.

- **Cyklus** - je netriviálny uzavretý ťah v ktorom sa okrem prvého a posledného vrchola žiaden neopakuje
- **Kružnica** - je súvislý pravidelný graf druhého stupňa
- **Acyklický graf** - je graf, ktorého podgraf neobsahuje kružnicu
- **Strom** - súvislý acyklycký graf
- Nasledujúce tvrdenia sú ekvivalentné:
  a) G = (V,H) je strom.
  b) V grafe G = (V,H) existuje pre každé u, v e V jediná u–v cesta.
  c) Graf G = (V,H) je súvislý a každá hrana množiny H je mostom.
  d) Graf G = (V,H) je súvislý a |H| = |V| − 1.
  e) V grafe G = (V,H) platí |H| = |V| − 1 a G je acyklický.
- Graf G=(V,H,k) budeme nazývať **koreňovým stromom**, kde k je pevne vybraný vrchol, ktorý nazývame koreň.
- **úroveň vrchola** v koreňovom strome je dĺžka - počet hrán jedinej k-u cesty
- **výška koreňového stromu** je maximálna úroveň vrchola zo všetkých úrovní vrcholov koreňového stromu
- prehľadávanie grafu G=(V,H) do hĺbky (**Depth-First Search**)
  - Nech strom T je triviálny strom obsahujúci 1 vrchol, polož p(v) := 1, k := 1
  - ak ešte T neobsahuje všetky vrcholy grafu, GOTO3, inak STOP
  - v grafe G so stromom T nájdi hraničnú hranu h = {u, v} s maximálnou značkou p(v) zaradeného vrchola u.
  - polož T := T [{h} [{v}, k := k + 1, p(v) := k, GOTO 2
- **algoritmus na prehľadávanie do šírky** je taký istý, len sa nehľadá maximálna značka ale minimálna (**Breadth-First Search**)

5. Kostra grafu, najlacnejšie a najdrahšia kostra grafu, Kruskalov algoritmus I. a II. Využitie pre hľadanie cesty maximálnej priepustnosti v grafe. Využitie Kruskalovho algoritmu na určenie komponentov grafu.

- **Kostra** súvislého grafu G=V,H je taký jeho faktorový podgraf, ktorý je stromom
- Keď G=V,H,c je hranovo ohodnotený graf a K je jeho kostra. Cena c(K) kostry je súčet ohodnotení jej hrán.
- **Najlacnejšia Kosta** v grafe G je kostra s najmenšou cenou
- **Najdrahšia kostra** v grafe G je kostra s najväčšou cenou
- na hľadanie najlacnejšej (najdrahšej) kostry používame **Kruskalov algoritmus I**:
  - zoradíme hrany podľa ich ohodnotenia vzostupne (zostupne) do postupnost P
  - nech prvá hrana postupnosti je {u,v}. Vylúč hranu u,v z postupnost P a ak už s vybranými hranami netvorí cyklus, zaraď ju do kostry.
  - ak je poćet hrán vybraných rovný V-1 alebo ak je postupnosť P prázdna - STOP, inak GOTO2
- na hľadanie najlacnejšej (najdrahšej) kostry používame **Kruskalov algoritmus II:**
  - Zorad hrany podla ich ohodnotenia vzostupne (zostupne) do postupnosti P.
  - Pre kazdy vrchol i e V poloz k(i) = i .
  - Nech prva hrana v postupnosti P je hrana {u, v}. Vyluc hranu {u, v} z postupnosti P. Ak k(u) 6= k(v), zarad hranu {u, v} do kostry, a lubovolne i e V, pre ktore k(i) = k(v), potom poloz k(i) := k(u)

Cesta maximálnej priepustnosti

- Nech G=V,H,c je hranovo ohodnotený graf, v ktorom hrany h patrí H c(h) > 0 znamená priepustnosť
- **priepustnosť** c(mikro(u,v)) u-v cesty (sledu, polosledu ...) mikro(u,v) definujeme ako: c(mikro(u,v)) = min { c(h) | h e mikro(u,v)}
- u,v cesta mikro(u,v) v grafe G=V,H,c je u-v **cesta maximálnej priepustnosti**, ak má najväčšiu priepustnosť zo vštkých u-v ciest v G

- **Algoritmus na hľadanie u-v cesty maximálnej priepustnosti** v súvislom hranovo ohodnotenom grafe G=V,H,c:
  - V grafe G zostroj najdrahšiu kostru K
  - V kostre K nájdi jedinú u-v cestu. Táto jediná u-v cesta v kostre K je u-v cestou maximálnej priepustnosti v grafe G
  - algoritmus nájde cestu maximálnej priepustnosti, ale táto cesta nie je z hľadiska prejdenej vzdialenosti optimálna
- **Algoritmus na hľadanie najkratšej u-v cesty s maximálnou priepustnosťou** v súvislom hranovo ohodnotenom grafe G=(V,H,c,d) kde c(h) je priepustnosť a d(h) je dĺžka hrany h patri H
  - v grafe G nájdi cestu maximálnej priepustnosti vzhľadom na ohodnotenia hrán c. Nech C je priepustnosť cesty mikro(u,v)
  - vytvor graf G'= (V,H', d) kde H'= {h|h patri H, c(h) >= C}. H' obsahuje len ie hrany ktoré majú priepustnosť väčšiu alebo rovnú než C.
  - V grafe G' nájdi najkratšiu u-v cestu vzhľadom na ohodnotenie hrán d
- Poznámka. Kruskalov algoritmus II (algoritmus 4.4) môžeme s výhodou použiť na zistenie komponentov grafu. Ak totiž spustíme tento algoritmus na nesúvislý graf, po skončení práce algoritmu značky k( ) vrcholov z V budú určovať komponenty nasledovne: Dva vrcholy u ∈ V , v ∈ V sú v tom istom komponente grafu G práve vtedy, keď k(u) = k(v) Pre zisťovanie komponentov grafu nie je potrebné usporiadať hrany podľa ich ohodnotenia a krok 1 algoritmu 4.4 možno preformulovať nasledovne: Zoraď hrany grafu G do postupnosti P v ľubovoľnom poradí.

6. Acyklické digrafy a ich vlastnosti. Typy súvislosti v digrafoch - orientovaná, neorientovaná a silná súvislosť. Monotónne očíslovanie vrcholov grafu. Algoritmy na hľadanie najkratšej a najdlhšej cesty v acyklických digrafoch

- **Acyklický digraf** je taký digraf, ktorý neobsahuje orientovaný cyklus
- **Strom** je suvisly acyklicky graf
- **Orientovaný strom** je neorientovane súvislý digraf, ktorý neobsahuje polocyklus
- Nasledujúce tvrdenia sú ekvivalentné:
  a) G = (V,H) je strom.
  b) V grafe G = (V,H) existuje pre každé u, v e V jediná u–v cesta.
  c) Graf G = (V,H) je súvislý a každá hrana množiny H je mostom.
  d) Graf G = (V,H) je súvislý a |H| = |V| − 1.
  e) V grafe G = (V,H) platí |H| = |V| − 1 a G je acyklický.
- Vlastnosti orientovaných stromov:
  - v digrafe G existuje pre každé u,v patri V jedina u-v polocesta
  - digraf G je neorientovane súvislý a každá orientovaná hrana H je mostom
  - digraf G je neorientovane súvislý a H = V - 1
  - v digrafe platí H = V -1 a G neobsahuje polocyklus
- Nech G = V,H je acyklický digraf. Potom V obsahuje aspoň jedn vrchol z taký že ideg(z) = 0 a apoň jeden vrchol u taký že odeg(u) = 0
- Očíslovanie vrcholov v1, v2, ... vn acyklického digrafu G = V,H platí: ak (vi, vk) patri H, potom i < k, nazveme monotónnym očíslovaním vrcholov acyklického digrafu
- **Monotónne očíslovanie acyklického digrafu**:
  - Krok 1. Polož i=1
  - Krok 2. Vezmi taký vrchol v ∈ V , že ideg(v) = 0 a polož vi := v.
  - Krok 3. Ak V - {v} = prázdna - STOP, inak G=G - {v}, i=i+1 a GOTO 2
- **Algoritmus na výpočet najkratšej u-v cesty v neorientovane súvislom acyklickom hranovo ohodnotenom digrafe** $\vec{G} = (V, H, c)$.
  - Krok 1. Monotónne očísluj vrcholy digrafu $\vec{G}$, nech P = v1, v2, . . . , vk postupnosť vrcholov digrafu $\vec{G}$ zoradená podľa monotónneho očíslovania. Zisti index vrchola u v postupnosti P. Nech i je index taký, že u = vi.
  - Krok 2. Pre každý vrchol v ∈ V priraď značky t(v), x(v). Polož t(u) := 0, t(j) := ∞ pre všetky j != u, j ∈ V . Polož x(j) := 0 pre všetky j ∈ V .

- Krok 3. Pre všetky vrcholy w výstupnej hviezdy vrchola vi také, že w != vi, urob:
  Ak t(w) > t(vi) + c(vi, w), potom t(w) = t(vi) + c(vi, w), a x(w) := vi.
- Krok 4. i := i + 1. Ak i = n STOP, inak GOTO Krok 3.

7. Časová analýza projektov - metóda CPM. Dve možné reprezentácie (vrcholovo ohodnoteným resp.- hranovo ohodnoteným digrafom). Najskôr možný začiatok vykonávania činnosti a najneskôr nutný koniec vykonávania činnosti. Trvanie projektu. Kritické činnosti a kritická cesta.

- Predpokladáme začiatok vykonávania projektu v čase 0. **Najskôr možný začiatok** Ti činností vychádzajúcich z vrchola i je prvý časový okamih, kedy skončí vykonávanie poslednej z činností vchádzajúcich do vrchola i.
- **Trvanie projektu Tn** je najskôr možný začiatok činností vychádzajúcich z konca projektu n.
- **Najneskôr nutný koniec** T′i činností vchádzajúcich do vrchola i je posledný časový okamih, po ktorý sa činnosti vchádzajúce do vrchola i môžu oneskoriť bez toho, aby sa zväčšilo trvanie projektu Tn.
- **Časová rezerva** Ri vo vrchole i je Ri = T′i − Ti.
- **Hodnotu Tn trvania projektu** určíme ako Tn = dmax(1, n). . t. j. dĺžku najdlhšej orientovanej cesty od začiatku projektu 1 do konca projektu n. Každú orientovanú cestu dĺžky trvania projektu Tn v sieťovom digrafe nazveme **kritickou cestou** (kritických ciest môže existovať aj viac).
- Činnosti ležiace na kritickej ceste sa nazývajú **kritické činnosti**.
- Algoritmus 5.6. **Algoritmus II. na určenie najskôr možných začiatkov z(v)** elementárnych činností v digrafe $-\!\!\rightarrow G\prec\prec = (V, H, c)$.
  - Krok 1. Vytvor monotónne očíslovanie v1, v2, . . . , vn vrcholov digrafu $-\!\!\rightarrow G\prec\prec$
  - Krok 2. Každému vrcholu v ∈ V priraď dve značky z(v), x(v).
    Pre každé v ∈ V inicializačne polož x(v) := 0, z(v) := 0.
  - Krok 3. Postupne pre k = 1,2, . . . , n − 1 urob:
    Pre všetky také vrcholy w výstupnej hviezdy vrchola vk, že w 6= vk, urob:
    Ak z(w) < z(vk) + c(vk), potom z(w) := z(vk) + c(vk) a x(w) := vk.
  - Krok 4. Vypočítaj trvanie projektu
    T := max{z(w) + c(w) | w ∈ V, odeg(w) = 0}
- **Algoritmus II. na určenie najneskôr nutných koncov k(v)** elementárnych činností v digrafe $-\!\!\rightarrow G\prec\prec = (V, H, c)$.
  - Krok 1. Vytvor monotónne očíslovanie v1, v2, . . . , vn vrcholov digrafu $-\!\!\rightarrow G\prec\prec$.
  - Krok 2. Každému vrcholu v ∈ V priraď dve značky k(v), y(v). Nech T je trvanie projektu. Pre každé v ∈ V inicializačne polož k(v) := T , y(v) := 0 .
  - Krok 3. Postupne pre i = n − 1, n − 2, . . . ,1 urob:
    Pre všetky vrcholy w výstupnej hviezdy vrchola vi také, že w 6= vi, urob:
    Ak k(vi) > k(w) − c(w), potom k(vi) := k(w) − c(w) a y(vi) := w

Pre každý vrchol i sieťového digrafu vypočítame Ti, t. j. najskôr možný začiatok činností vychádzajúcich z vrchola i, ako Ti = dmax(1, i) a T′i najneskôr nutný koniec činností vchádzajúcich do vrchola i ako T′i = Tn − dmax(i, n) kde dmax(x, y) je dĺžka najdlhšej orientovanej x–y cesty. Pre vrchol i ležiaci na nejakej kritickej ceste je Ti = T′ i, pre ostatné vrcholy je Ti < T ′i.

8. Eulerovský ťah v grafe. Eulerovský graf. Kritérium pre to, aby bol graf eulerovský. Algoritmy na zostrojenie uzavretého eulerovského ťahu v grafe (Fleuryho, labyritnový, postupným rozširovaním uzavretého ťahu).

- hovorıme, ze sled s(u, v) v suvislom grafe G = (V,H) je eulerovsky,ak obsahuje vsetky hrany grafu G.
- Kedze tah obsahuje kazdu hranu grafu G prave raz, postupnost vrcholov a hran tahu t(u, v) predstavuje postup, ako nakreslit diagram grafu G ”jednym tahom”.
- hovorıme, ze graf G = (V,H) je eulerovsky, ak v nom existuje uzavrety eulerovsky tah.
- Suvisly graf G = (V,H) je eulerovsky prave vtedy, ked stupne vsetkych vrcholov grafu G su parne.
- Algoritmus na konstrukciu eulerovskeho tahu :
    - Zacni z lubovolneho vrchola z, poloz T = (z) a postupne predlzuj tah T pokial sa da. Ukoncıs vo vrchole z.
    - Najdi prvy vrchol v tahu T , ktory ma este aspon jednu hranu nepouzitu v tahu T . Ak taky vrchol v neexistuje, STOP.
      Tah T je hladanym uzavretym eulerovskym tahom.
    - Vytvor tah S takto: Poloz S = (v) a postupne predlzuj tah S doteraz nepouzitymi hranami, pokial sa da. Ukoncıs vo vrchole v.
    - Rozdel tah T na z–v tah T1 a v–z tah T2, t. j. T = T1 + T2. Poloz T = T1 + S + T2. GOTO Krok 2.
- Fleuryho algoritmus na hladanie eulerovskeho tahu:
    - Zacni v lubovolnom vrchole a do tahu T zarad lubovolnu s nım incidentnu hranu.
    - Ak su do tahu T zaradene vsetky hrany grafu G, STOP.
    - Ako dalsiu hranu zarad do tahu T taku hranu incidentnu s jeho poslednym vrcholom, po vybratı ktorej sa podgraf grafu G pozostavajuci z nevybratych hran a s nimi incidentnych vrcholov nerozpadne na
      - dva netrivialne komponenty
      - netrivialny komponent a izolovany zaciatok tahu T .
    - GOTO krok 2.
- Labyrintovy algoritmus na hladanie uzavreteho eulerovskeho tahu:
    - Zacni z lubovolneho vrchola u e V. Nech sled S inicializacne pozostava z jedineho vrchola u. Poloz w := u – vrchol w je posledny vrchol doteraz vytvoreneho sledu S.
    - Ako dalsiu hranu vyber podla nizsie uvedenych pravidiel do sledu S hranu {w, v}. Zaznac si smer pouzitia hrany {w, v}. Ak doteraz vrchol v este nebol zaradeny do sledu S, oznac hranu {w, v} ako hranu prveho prıchodu.
      Dalej zaznamenaj tzv. sp¨atnu postupnost — poradie hran, v ktorom sa v slede S vyskytuju po druhykrat. Pri vybere hrany dodrzuj nasledujuce pravidla:
      (L1): Kazdu hranu mozno v jednom smere pouzit iba raz
      (L2): Poradie zaradovania hran:
      – nepouzite hrany
      – hrany pouzite raz
      – hrana prveho prıchodu (ak niet inej moznosti)
    - Ak taka hrana neexistuje – STOP.
      Spatna postupnost urcuje hladany eulerovsky tah.
    - Inak poloz w := v a pokracuj krokom 2.

9. Úloha čínskeho poštára. Párenie v grafe. Edmondsov algoritmus na riešenie úlohy čínskeho poštára.

- slovne: Poštár má výjsť z pošty, prejsť všetky ulice svojho regiónu a vratíť sa späť tak aby sa čo najmenej nachodil
- matematicky: V súvislom hranovo ohodnotenom grafe nájsť uzavretý eulerov sled najmenšej dĺžky
- ak má graf vrcholy s párnym stupňom, stačí zostrojiť uzavretý eulerov ťah
- ak má vrcholy s nepárnym stupňom, potom ich tam je párny počet 2t
- pridaním fiktívnych hrán typu {nepárny, nepárny} s dĺžkou rovnajúcou sa vzdialenosti príslušných vrccholov v G možno dostať eulerovský graf
- čím menší súčet dĺžok pridaných fiktívnych hrán, tým lepšie riešenie
- eulerovský sled: sled s(u,v) v súvislom grafe G, ak obsahuje všetky hrany grafu G
- eulerov ťah je taký ťah t(u,v) v súvislom grafe G, ktorý obsahuje všetky hrany grafu G
- eulerov graf je taký graf, v ktorom existuje uzavretý eulerovský ťah
- párenie je taký podgraf P grafu, ktorého každý stupeň vrchola je práve 1
- cena párenia je súčet ohodnotení hrán párenia
- maximálne párenie - hovoríme, že párenie P je maximálne párenie v grafe G, ak P nie je podgrafom žiadneho iného párenia v G
- najpočetnejšie párenie - párenie P je najpočetnejšie párenie v grafe G ak P má zo všetkých párení najväčší počet hrán
- úplné párenie je párenie P, ktoré obsahuje všetky vrcholy pôvodného grafu G (P je faktorový podgraf grafu G)
- na riešenie úlohy používame Edmondsov algoritmus:
    - nájdeme vrcholy s nepárnym počtom stupňov - párny počet
    - z týchto vrcholov spravíme úplný graf G' a ohodnotíme vzdialenosťami koncovych vrcholov hrany v G
    - v grafe G' spravíme úplné párenie s najnižšou cenou
    - hrany párenie pridáme do hranovej mnoziny pôvodného grafu - dostaneme multigraf, v ktorom majú všetky vrcholy párny počet stupňov
    - zostrojíme najkratší eulerovský tah T
    - hrany párenia v ťahu T nahraď najkratšími cestami v G a označ ich ako prejdené na prázdno. Dostaneme najkratší eulerovský uzavretý sled v G

10. Úloha obchodného cestujúceho. Hamiltonovský cyklus a hamiltonovský graf. Postačujúce podmienky pre to, aby graf bol hamiltonovský. Metóda zdvojenia kostry a metóda kostry a párenia. Vytváracie a zlepšujúce heuristiky.

- slovne: Obchodný cestujúci navštíviť všetkých svojich zákazníkov a vrátiť sa domov tak, aby sa čo najmenej nachodil
- matematicky:
    - ak dovolujeme navštíviť viackrát: V súvislom hranovo ohodnotenom grafe nájsť najkratší uzavretý hamiltonovský sled. Hamiltovovský sled je taký sled s(u,v), ktorý obsahuje každý vrchol grafu G.
    - ak nedovoľujeme: V súvislom hranovo ohodnotenom grafe G hľadáme najkratší hamiltovovský cyklus. Hamiltonovský cyklus je špeciálny prípad hamiltonovského sledu.
- hamiltonovský graf je taký graf, ktorý obsahuje hamiltonovský cyklus
- v praxi sa hamiltonovský cyklus vyskytuje veľmi málo a tak sa sústredíme na hľadanie hamiltonovského sledu. Zakazovať návštvy aj tak nemá zmysel.
- ak v grafe hamiltonovský cyklus neexistuje hľadáme ho pomocou hamiltonovského sledu v úplnom grafe G', ohodnotenom vzdialenosťami v pôvodnom grafe.
- v grafe G' platí trojuholníková nerovnosť d(u,v) <= d(u,w) + d(w,v)
- na riešenie máme suboptimálne algoritmy, ktoré dávajú dostatočne dobré riešenia ale nie dostatočne efektívne

- Greedy Method (pažravá metóda)
    - Zacni v lubovol'nom vrchole a do (buduceho) hamiltonovskeho cyklu vloz najlacnejsiu hranu incidentnu s tymto vrcholom.
    - ak je vybraných n-1 hrán, uzavri cyklus
    - Inak vyber taku najlacnejsiu nevybranu hranu incidentnu s poslednym vrcholom doteraz vybranej postupnosti, ktora nie je incidentna so ziadnym inym vrcholom vybranej postupnosti. Goto2
    - každú hrany cyklu nahradíme príslušnou najkratšou cestou v pôvodnom grafe G
- Metóda zdvojenia kostry
    - v grafe G zostrojíme najlacnejšiu kostru
    - v kostre K zostrojíme uzavretý sled S, ktorý obsahuje každú hranu práve 2x. (použijeme napríklad Tarryho algoritmus)
    - z uzavretého sledy vytvoríme hamiltonovský cyklus takto: prechádzaj sledom S a keď narazíš na vrchol alebo úsek vrcholov za sebou, ktoré sa opakujú, premosti ich priamou čiarou.
- Algoritmus kostry a párenia
    - v grafe G zostroj najlacnejšiu kostru K
    - v kostre K nájdi všetky vrcholy nepárneho stupňa - párny počet 2t
    - z týchto vrcholov zostroj úplný graf G' - jeho hrany ohodnoť podľa pôvodného grafu
    - v grafe G' najdi úplné párenie s minimálnou cenou
    - hrany párenia pridaj do najlacnejšej kostry K. Dostanes multigraf s párnym stupňom vrcholov
    - v grafe (multigrafe) zostroj uzavretý eulerovský ťah T
    - hamiltonovský cyklus takto: prechádzaj ťahom T a ked narazíš na opakujúce sa vrcholy alebo úsek vrcholov, premosti ich priamou hranou
- Vytvárajúce heuristiky skonštruujú nový hamiltonovský cyklus. Zlepšujúce heuristiky sa snažiavylepšiť súčasný hamiltonovský cyklus.

11. Algoritmy a ich zložitosť. Definícia symbolu $O(f(n))$, polynomiálne algoritmy. Polynomiálna redukcia a polynomiálna transformácia. Polynomiálne riešiteľné úlohy a NP-ťažké úlohy.

- Pod pojmom algoritmus rozumieme postupnosť krokov, ktorá nás dovedie k žiadanému riešeniu daného problému. Žiada sa, aby algoritmus mal tieto vlastnosti:
    - determinovanosť – má byť zadaný konečným počtom jednoznačných pravidiel
    - efektívnosť – má zaručiť vyriešenie úlohy po konečnom počte krokov
    - hromadnosť – má byť použiteľný na celú triedu prípadov úlohy svojho typu
- Pre ohodnotenie výpočtovej zložitosti algoritmu nás však viac ako jeden konkrétny prípad zaujíma závislosť počtu elementárnych krokov algoritmu na veľkosti resp. rozsahu počítanej úlohy. Budeme definovať dĺžku úlohy ako množstvo vstupných dát príslušnej úlohy.

Príklad
Pre graf G = (V, H) alebo digraf $\vec{G} = (V, H)$ s n vrcholmi a m hranami, t.j. kde |V| = n, |V| = m, bude dĺžka úlohy (m + n). Niekedy sa udáva len závislosť' času výpočtu len na počte vrcholov n, pričom sa berie do úvahy, že $m \leq n(n-1)/2 \leq n2$ pre grafy, resp. $m \leq n(n-1) \leq n2$ pre digrafy.

- Počet krokov algoritmu môže totiž závisieť nielen od množstva vstupných dát, ale aj od ich vzájomnej konfigurácie. Preto funkcia T (n) môže vyjadrovať iba hornú hranicu počtu krokov algoritmu pre najhorší prípad úlohy s dĺžkou n (worst case analysis).
- Nech g(n), h(n) sú dve kladné funkcie definované na množine prirodzených čísel. Budeme písať g(n) = O(h(n)) a hovoriť, že funkcia h(n) **asymptoticky dominuje funkcii** g(n), ak existuje konštanta K a prirodzené číslo n0 také, že $g(n) \leq K.h(n)\ \forall n \geq n_0$.
  Hovoríme, že **algoritmus A má zložitosť'** O(f(n)), ak pre horný odhad T (n) počtu krokov algoritmu A pre úlohu dĺžky n platí T (n) = O(f(n)).

Skrátene hovoríme o O(f(n)) algoritme. Špeciálne ak $f(n) \leq n^k$ pre nejaké konštantné k, hovoríme, že A je **polynomiálny algoritmus**.
Hovoríme, že problém má zložitosť nanajvýš O(f(n)), ak preň existuje O(f(n)) algoritmus.

- Nech je daný problém P_1 so vstupnými dátami D_1. Problém P_1 môžeme riešiť aj tak, že ho pretransformujeme na iný problém P_2 tak, že dáta D_1 prerobíme na dáta D_2 problému P_2. Ak riešenie R_2 problému P_2 vieme prerobiť na riešenie R_1 problému P_1, transformácia je hotová. Ak prerobenie dát D_1 na D_2 a riešenia R_2 na R_1 vyžaduje len polynomiálny počet elementárnych operácií, nazveme túto transformáciu **polynomiálnou transformáciou**.
- Ak prevody dát a riešení vyžadujú len polynomiálny počet elementárnych operácií a výpočet vyžaduje polynomiálny počet výpočtov problému P2, hovoríme, že sme problém P1 **polynomiálne redukovali na problém** P2. Ak problém P1 možno polynomiálne redukovať na problém P2 a naopak, problém P2 možno polynomiálne redukovať na P1 hovoríme, že problémy P1, P2 sú **polynomiálne ekvivalentné**.
- Ako etalón ťažkých úloh bola vybratá úloha bivalentného lineárneho programovania (BLP). Problém, ktorý možno polynomiálne redukovať na úlohu BLP, je ľahší alebo rovnako ťažký ako úloha BLP. Problém, na ktorý možno polynomiálne redukovať úlohu BLP je ťažší alebo rovnako ťažký ako úloha BLP.
- Hovoríme, že **problém P je NP–ťažký**, ak úlohu bivalentného lineárneho programovania možno polynomiálne redukovať na P. Hovoríme, že **problém P je NP–ľahký**, ak problém P možno polynomiálne redukovať na úlohu bivalentného lineárneho programovania. Hovoríme, že **problém P je NP–ekvivalentný**, ak je problém P polynomiálne ekvivalentný s úlohou bivalentného lineárneho programovania.

12. Alokačné úlohy - depá a havarijné stradiské. Vážený p-medián a vážené p-centrum. Heuristický výmenný algoriotmus. Atrakčné obvody.

- Pri zásobovaní územia nejakým tovarom (uhlím, nábytkom, potravinami, liekmi atď.) často potrebujeme rozhodnúť, koľko skladov a v ktorých lokalitách máme postaviť. Podobný problém môžeme riešiť pri rozhodovaní koľko stredísk zdravotnej záchrannej služby a v ktorých miestach zriadiť. Tieto problémy spadajú pod tzv. lokačné problémy. Lokačné problémy sa líšia typom kriteriálnej funkcie a modelom prostredia, v ktorom ich riešime. Existuje niekoľko spôsobov na modelovanie a riešenie lokačných problémov. Najdôležitejšími z nich sú metódy celočíselného lineárneho programovania a metódy teórie grafov. My sa, pochopiteľne, budeme zaoberať metódami teórie grafov v najjednoduchšom prípade, kedy je už množstvo skladov či záchranných stredísk známe. Modelom prostredia, v ktorom budeme tieto úlohy riešiť, bude súvislý hranovo a vrcholovo ohodnotený graf G = (V, H, c, w), v ktorom vrcholy predstavujú križovatky a dôležité body, hrany modelujú ulice, resp. priame cesty medzi vrcholmi, ohodnotenie c(h) hrany $h \in H$ predstavuje dĺžku hrany h, ohodnotenia w(v) vrchola $v \in V$ – váha vrchola v (predstavuje relatívnu dôležitosť vrchola v). Všetky vrcholy v grafe G = (V, H, c, w) potrebujú obsluhu. Ich náročnosť na obsluhu je vyjadrená ich váhou. Niektoré vrcholy v grafe G môžu naviac slúžiť ako strediská obsluhy. Poznáme dve základné funkcie stredísk obsluhy. Prvá z nich je funkcia zásobovacia. V tomto prípade pre stredisko obsluhy používame termín depo. V depe je umiestnený sklad materiálu. Každý vrchol v grafu G = (V, H, c, w) potrebuje za jednotku času w(v) jednotiek materiálu, jednotkové náklady na dovoz materiálu sú úmerné prepravovanej vzdialenosti. Tu hľadáme také umiestnenie diep, ktoré minimalizuje celkové dopravné náklady na obsluhu všetkých vrcholov grafy G. Druhá funkcia stredísk obsluhy je záchranná. Takú funkciu plnia napríklad stanice pohotovostnej lekárskej služby, požiarne zbrojnice, strediská horskej služby atď. V tomto prípade pre stredisko obsluhy používame termín havarijné stredisko. Tu už dopravné náklady nehrajú takú dôležitú úlohu ako v predchádzajúcom prípade — kritériom je tu dostupnosť najhoršie položeného vrchola grafu G = (V, H, c, w). Chceme nájsť umiestnenia staníc záchrannej lekárskej služby tak, aby ani ten najhoršie položený pacient neumrel pre neskorý prí chod pomoci, chceme nájsť umiestnenie požiarnych zbrojníc tak, aby v prípade potreby požiarnici došli včas, aj keby požiar vznikol aj v najhoršie položenom mieste.
- Nech $1 \leq p < |V|$, Dp p-prvková podmnožina množiny V . Hovoríme, že Dp je vážený p-medián grafu G, ak pre ľubovoľnú p-prvkovú podmnožinu D′p množiny V platí $f(Dp) \leq f(D'p)$, t.j. ak súhrnn vážená vzdialenosť všetkých vrcholov grafu G od Dp je najmenšia medzi

všetkými p-prvkovými podmnožinami množiny V . Špeciálne ak w(v) = 1 pre všetky v ∈ V , hovoríme, že Dp je p-medián.

- Nech $1 \leq p < |V|$, Dp p-prvková podmnožina množiny V .Hovoríme, že Dp je vážené p-centrum grafu G, ak pre ľubovoľnú p-prvkovú podmnožinu D′p množiny V platí $ecc(Dp) \leq ecc(D'p)$,t. j. ak množina Dp má najmenšiu váženú excentricitu zo všetkých p-prvkových podmnožín množiny V . Špeciálne ak w(v) = 1 pre všetky v ∈ V , hovoríme, žeDp je p-centrum.
- Heuristický algoritmus na hľadanie váženého p-mediánu v súvislom hranovo a vrcholovo ohodnotenom grafe G = (V, H, c, w) .
  - Krok 1. Náhodne vyber p-prvkovú podmnožinu množiny V .
    Nech Dp = {v1, v2, . . . , vp}, V − Dp = {u1, u2, . . . , uq}, kde q = |V | − p.
  - Krok 2. Hľadaj také i, j, $1 \leq i \leq p$, $1 \leq j \leq q$,
    že pre D′p(i, j) = (Dp ∪ {uj}) − {vi} je f(D′p) < f(Dp).
  - Krok 3. Ak taká dvojica indexov i, j neexistuje, STOP.
    Inak polož Dp := D′p(i, j) a GOTO Krok 2.
- Nech je daný súvislý hranovo a vrcholovo ohodnotený grafG = (V, H, c, w) a p-prvková množina diep Dp. Atrakčný obvod A(v) depa v ∈ Dp je množina všetkých takých vrcholov grafu G, ktorých vzdialenosť od depa v je menšia alebo rovná ako vzdialenosť od iných diep, t.j. $A(v) = \{x | x \in V, \forall u \in Dp\ d(v, x) \leq d(u, x)\}$
- Prvotný atrakčný obvod A′(v) depa v ∈ Dp je množina všetkých takých vrcholov grafu G, ktorých vzdialenosť od depa v je menšia ako vzdialenosť od iných diep, t.j.
  $A'(v) = \{x | x \in V, \forall u \in Dp, u \neq v\ d(v, x) < d(u, x)\}$

13. Toky v sieťach. Dopravná sieť, zdroj, ústie. Tok -- definícia, veľkosť toku, maximálny tok. Rezová množina, Ford-Fulkersonova veta a algoritmus na hľadanie maximálneho toku v sieti. Algoritmus na hľadanie maximálneho toku s minimálnou cenou. Siete s viacerými zdrojmi a viacerými ústiami.

- sieťou nazveme neorientovane súvislý hranovo ohodnotený digraf G zo sipkou =(V,H,c) v ktorom ohodnotenie c(h) > 0 každej hrany h patrí H je celočíselné a predstavuje priepustnosť hrany h a v ktorom existuje
  - práve jeden vrchol z taky ze ideg(z) = 0, – zdroj
  - práve jeden vrchol odeg(u) = 0, - ústie
- pre každý vrchol digrafu platí značenie:
  - H+(v) - množina hrán z vrchola v vychádzajúcich
  - H-(v) je množina hrán do vrchola v vhcádzajúcich
- Tok v sieti G=(V,H,c) je celočíselná funkcia y: H->R definovaná na množine orientovaných hrán H, kde platí:
  - y(h) >= 0 pre vsetky h patri H
  - y(h) <= c(h) pre vsetky h patri H
  - suma (h patri H+(v)) y(h) = suma (h patri H-(v)) y(h) pre všetky v patri V okrem v=u a v=z
  - suma (h patri H+(z)) y(h) = suma (h patri H-(u)) y(h)
- Veľkosťou toku y nazveme číslo F(y) = suma (h patri H+(v)) y(h), čo sa rovná suma (h patri H-(u)) y(h)
- tok v sieti G je maximálny, ak má najvaäčšiu veľkosť zo všetkých možných tokov v sieti G
- Orientovanú hranu nazveme nastenou, ak c(h) = y(h)
- rezerva hrany v poloceste mikro(v,w): r(h) = c(h) - y(h) ak je hrana h použitá v smere orientácie, r(h) = y(h) ak je hrana použitá proti smeru orientácie polocesty
- Rezerva polocesty mikro(u,v) je minimum rezer hrán tejto polocesty
- polocesta je rezervná polocesta ak má kladnú rezervu
- rezervná polocesta mikro(u,v) zo zdroja do ústia sa nazýva zväčšujúca polocesta
- Ford-Fulkerson veta: Toky v sieti G=V,H,c so zdrojom z a ústím u je maximálny práve vtedy keď neexistuje z-u zväčšujúca polocesta
- Ford-Fulkerson algoritmus na hľadanie maximálneho toku v sieti:
  - Zvoľ v sieti začiatočný tok y, napríklad nulový tok

- nájdi v sieti G s tokom y zvačšujúcu polocestu mikro(z,u)
- ak zväčšujúca polocesta neexistuje, tok y je maximlny - STOP
- ak zväčšujúca polocesta mikro(z,u) existuje a má rezervu r zmeň tok y nasledujúco:
  - y(h) = y(h) ak h nele69 na ceste mikro}z,u|
  - y(h) = y(h) + r ak h leží na ceste v smere svojej orientácie
  - y(h) = y(h) - r ak h leží na ceste proti smeru svojej orientácie
  - GOTO2

14. Rovinné grafy. Stena rovinného diagramu, Eulerov vzorec. Maximum počtu hrán rovinného grafu. Homeomorfizmus grafov. Prototypy najjednoduchších nerovinných grafov. Kuratowského veta.

- Graf je usporiadaná dvojica G = (V,H), kde V je neprázdn konečná množina a H je množina neusporiadaných dvojíc typu {u,v} takých že u patri V, v patri V a u!=v.
- Prvky množiny V nazývame vrcholy a prvky množiny H hranami grafu G.
- Digraf - usporiadnaá dvojica - (u,v) - V - vrcholy, H - orientované hrany digrafu
- Diagram grafu je grafická reprezentácia grafu - jeho príslušný obrázok
- graf G=V,H nazveme rovinný, ak k nemu existuje rovinný diagram
- Diagram grafu nazveme rovinný ak sa jeho hrany nepretínajú nikde inde okrem vrcholov
- niekde aj planárny graf, nákresy ...
- stena je maximálna časť roviny, ktorej 2 ľubovoľné body možno spojiť súvislou čiarou nepretinajúcou žiadnu hranu rovinného diagramu.
- 2 druhy stien, vonkajšia - práve jedna stena je neohraničená a ďalšie sú vnútorné
- Eulerova polyedrická formula: Nech G=V,H je súvislý rovinný graf a S je množina stien jeho rovinného diagramu, potom platí: S = H - V + 2
- Maximum počtu hrán rovinného grafu: Nech G=V,H je mximálny rovinný graf s množinou vrcholov V, kde V >= 3. Potom: H = 3*V - 6, teda H<=3*V-6
- úplný graf s piatimi vrcholmi K5 a úplný bipartitný graf K3,3 nie sú rovinné
- Rozpoltenie hrany - rozdelenie hrany na 2 hrany a v bode rozdelenia pridáme vrchol
- grafy G=V,H a G'=V',H' sú homeomorfné ak sú izomorfné alebo ak konečným počtom rozpoľtení hrán môžeme dostať izomorfné grafy
- Kuratowski - graf G je rovinný práve vtedy ak ako podgraf neobsahuje graf homeomorfný s K5 alebo K3,3

15. Farbenie grafu, n-zafarbiteľnosť grafu, chromatické číslo grafu. Heuristiky na farbenie grafov. Praktické úlohy vedúce na riešenia úlohy farbenia grafu.

- môžeme ilustrovať na úlohe zafarbenia štátov na politickej mape tak, aby žiadne 2 susedné štáty neboli zafarbené rovnakou farbou.
  - každému štátu i moru pridelíme jeden vrchol
  - vrcholy pospájame susedný so susedným
  - diagram výsledného grafu
- **zafarbenie grafu** je funkcia ktorá každému vrcholu priradí práve jednu farbu
- **prípustným zafarbením** nazveme také zafarbenie, ktoré žiadnym 2 susedným vrcholom nepriradí tú istú farbu
- graf G = (V,H) nazveme **k-zafarbiteľným** ak jeho vrcholy možno prípustne zafarbiť k farbami (t. j. tak, aby ziadne dva susedne vrcholy neboli zafarbene rovnakou farbou.)
- **chromatické číslo grafu** je najmenšie prirodzené číslo **k** také, že graf G je k-zafarbiteľný. Chromatické číslo budeme značiť symbolom **X(G)**
- **Sekvenčné farbenie grafu:**
  - Nech P = v1, v2 ... vn je ľubovolná postupnosť vrcholov grafu
  - Postupne pre i=1,2 ...n urob: Zafarbi vrchol vi farbou najmenšieho čísla takou, že žiaden zo zafarbených susedov vrchola vi nie je zafarbený touto farbou.
  - algoritmus potrebuje najviac max {deg(v) | v patri V} + 1 farieb, resp (G) 1 + max{deg(v) | v 2 V}
- **Paralelné farbenie grafu:**

- Zoraď vrcholy G do postupnosti P podľa stupňa vrchola nerastúco. Inicializuj množinu farieb F = {1}; j=1;
- Postupne prechádzaj vrcholy vi postupnosti P a ak vrchol vi nemá suseda zafarbeného farbou j, tak ho farbou j zafarbi.
- ak sú všetkcy vrcholy postupnosti zafarbené, STOP
- a nie sú zvýš počet farieb, j=j+1; F=F zjednotenie {j} a goto2

- **Aplikácie**: priraďovanie rádiových frekvencií, minimalizácia počtu fáz na svetelnej križovatke, minimalizácia nákupných tašiek....