

J. Kolář , K201

skouška hromadné termíny pro všechny

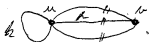
<http://pitbul.wz.cz>

Neorientovaný
graf - matem. struktura

- vršky

- hrany

- incidence - vztahy, které každé hraně
přikazuje neuspořádanou dvojici vrškových

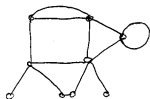


$$\varphi(e) = [u, v] = [v, u]$$

rovnoběžné hrany $\Rightarrow$ multigraf,

jinak graf prostý

smýčka $\varphi(e) = [u, u]$



multigraf

obecný g. - nemá rovnoběžné hrany ani smýčky
úplný g. - mezi každou dvojicí vrškových
vršek existuje hrana
značí se $K_n = \langle \binom{V}{2}, U \rangle$

K_3



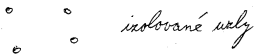
K_4



K_5



diskrétní graf - má uzly, nemá hrany



podgraf

graf G' podgraf G - uzly U' podmnožina U

- hrany H' $\subseteq H$

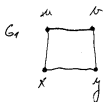
- incidence pro všechny H' stejná jako pro H

faktor grafu

podgraf se všemi uzly

faktorů grafu K_n $2^{\binom{n}{2}}$ pro $n \geq 2$

- sjednocení grafů - sjednocení uzlů, hran, incidence
nemůžeme se vychýlovat stejné hrany
s různými uzly,
jako by byly podgrafy nějakého supergrafu



$G_1 \cup G_2$



$G_1 \cap G_2$



disjunktní g. - nic společného
 hranově disj. - nemají společné hrany

doplňák G $G_1 \subseteq G$



nejmenší G_2 : $G_1 \cup G_2 = G$

$$G_2 = G - G_1$$

bez udání dalšího uvažujeme doplňák ke K_n ,
 který má stejnou množ. vrcholů jako G_1



doplňák ke G_1



rozdíl lib. 2 grafů

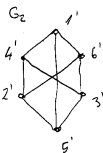
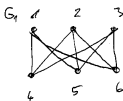
$$G_1 - G_2 = (G_1 \cup G_2) - G_2 = G_1 - (G_2 \cap G_1)$$

symetrická diference

$$G_1 \oplus G_2 = (G_1 \cup G_2) - (G_1 \cap G_2)$$

konečný graf - konečné hrany a vrcholy
 nekonečný - alespoň jedna množ. nekonečná

izomorfismus grafů - mají stejnou strukturu



$$i \leftrightarrow i'$$

(2)

stupeň uzlu u je počet hran, které s ním incidentují - $\delta(u)$

$$\sum_{u \in V} \delta(u) = 2|H|$$

soubor stupňů grafu - (nerostoucí) posloupnost sestavená ze stupňů všech uzlů grafu

př. $(5, 4, 4, 4, 3, 3, 3, 2, 2, 2)$

2 izomorfní grafy mají stejný soubor stupňů

V každém grafu je právě sudý počet uzlů lichého st.

Pravidelný graf - všechny uzly stejný stupeň

Sled s krajními uzly u, v - posloupnost uzlů a hran na libovolné hrane u do v
délka sledu - počet projitých hran

Trh - sled, ve kterém se každá hrana neopakuje

Cesta - trh, ve kterém se neopakuje žádný uzel

Uzavřený sled sled, ve kterém první a poslední uzel splývají

Kružnice - uzavřený sled, kde se neopakuje žádná hrana

Z každého sledu lze vybrat cestu

* konvexní grafu existuje sled mezi lib. dvěma uzly,
tedy i cesta

Komponenta grafu

př. graf s 5 komponentami



3

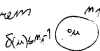
Uzavřený sled - má minimálně jednu hranu
samostatný uzel - * něj do něj je otevřený sled

1. V. Necht G je obecný graf

$$\delta(u) + \delta(v) \geq n - 1$$

složit T1 2/4

2. sporum



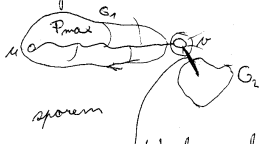
$$m = |U|$$

$$m_1 + m_2 \leq m$$

$$\delta(u) + \delta(v) \leq (m_1 + m_2) - 2 \leq m - 2$$

V: dajst T12/5 $U(G) \geq 2$, $\text{ešat nējmēnē 2 nēlji}$

D:



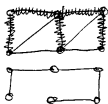
o šūda kramu bē šā prodūētā P_{max} ,
šūdā nēmokla bē maksimāls

Minimālā pācē kram sūvīstēko grafā

$$|H| \geq |V| - 1$$

Sakojmē grafm jē štrām - šēpē nēobšahjē kramīcē

Koštrā grafā - šāllor, kērj jē štrāmē



6 nēlji
→ 5 kram

graf - minimum kram $\frac{n-1}{2}$
maximum kram $\frac{n(n-1)}{2}$

graf p komponent, n nēlji - n-p kram min

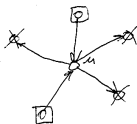
Orientované grafy

a hran mají šípku

hrana má počáteční a koncový uzel

incidence přirozeně uspořádanou dvojici uzlů

následník x předchůdce



□ předchůdce

× následník

rovnoběžné hrany - mají i stejnou orientaci



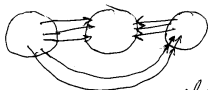
$2^{141} = 2^7$ různých možností orientace
a každá grafů třeba smýšlet že orientovat jedním způsobem

orientované spojení

orient. sah. $\rightarrow$ nepokryje se hrany

orient. cesta $\rightarrow$ nepokryje se uzly

silně souvislý graf - pro libovolnou dvojici ^{uzlů} existuje
 orientované spojení mezi nimi
 silné komponenty

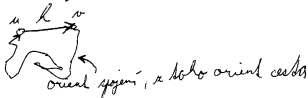


všechny hrany mezi sil. komp. musí
 být jednotným směrem
 nemusí být vidět, že jsou silnými komp.

V: složit $\Pi^3/3$

2: $1 \Rightarrow 2$ ✓

G silně souvislý $\Rightarrow$ souvislý



2 $\Rightarrow$ 3

G - souvislý, $\forall h$ k němu nějakým způsobem

$$\Rightarrow U = U_1 \cup U_2$$



musí být hrana vedoucí zpět,
 aby platilo

$3 \Rightarrow 1$

$$\nexists u_1, u_2 \quad \exists \begin{array}{c} u_1 \quad u_2 \\ \xrightarrow{\quad} \\ \xleftarrow{\quad} \end{array}$$

$\stackrel{?}{\Rightarrow} G \text{ je silne souvislý} \Leftrightarrow \nexists u, v \exists \mu \rightarrow \nu$

sporem $\mu, \nu: \mu \rightarrow \nu$

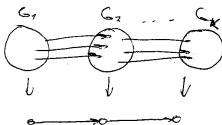


$$U_1 = \{ u \in U, \mu \rightarrow \nu \}$$

$$U_2 = U - U_1$$

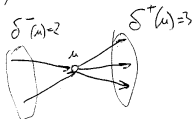
existí μ, ν $\mu \rightarrow \nu$ dostal i do datých,
spor

Kondenzace O.G.



Kondenzace
představuje situaci mezi
silnými komponentami

odstupní stupeň $\delta^-(u)$
výstupní stupeň $\delta^+(u)$



acyklický graf - nemá žádné cykly

$V_1: T13/5$

$D: \forall n: \delta^+(n) \geq 1$



na cyklu nemohou být uzly, které mají
je hrany výstupní / vstupní (hořez / listy),
by lze odebrat bez ohledu na narušení
možných cyklů

Topologické uspořádání

uzly očíslovány 1..n, aby všechny hrany
byly orientovány o uzlu s nižším číslu
s vyšším číslem
nelze u grafu s cykly

Reprezentace grafu

↙ maticová
spojnicová

Matice incidence NG

rad tělesem mod 2 - násobení a sčítání mod 2
 → stačí prvky 0/1 a lze s matricí provádět všechny běžné operace

smýšly - hrana má jen 1 ještníku - vadi rovnoběž. hr. - stejné sloupce

izomorfní grafy - wírké matice stejné velikosti
 nebo snadno wírké, časové složitě

hodnot

$k(A) \leq |V| - 1$; = pro souvislé grafy

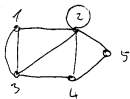
$k(A) \leq |V| - p$ pro graf s p. komponentami

Matice sousednosti NG

$V = [v_1, \dots, v_n]$

$a_{ij} = |E^{-1}(v_i, v_j)|$ množnost

počet hran mezi wly

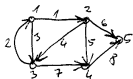


	1	2	3	4	5
1		1	2		
2	1		1	1	1
3	2	1		1	
4		1	1		1
5		1		1	

je matice symetrická

$V^T = [v_{jk}^{(n)}]$ počet sledů delky n mezi u_i a u_k
 $A \cdot A^T = V + D$, $D = [d_{ij}]$, kde $d_{ij} = \delta(u_i)$

Matice incidence OG



$A =$

	1	2	3	4	5	6	7	8
1	1	-1	1					
2	-1				1	1		
3		1	-1	-1			1	
4					-1		-1	1
5						-1	-1	

Matice souslednosti OG

$V =$

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



$V^* = \sum V^i$, $i=0, \dots, d$ kde $d = \min(|H|, |V|-1)$

matice doslednosti $R = [r_{ij}]$, $r_{ij} = 1$ pokud $u_i \rightarrow u_j$ jinak 0
 bylo by to časově složité

možná se nepokládá
 uhlý i hran

Spojová reprezentace grafu

NG - seznamy sousedů

OG - seznamy následníků

Prohledávání grafu - chceme projít všemi uzly a hranami

Prohledávání grafu do šířky

BFS - Breadth-First Search

Stavy uzlů

FRESH - ještě neprojitý

OPEN - procházíme

CLOSED - projitý a už jíme za ním

Dat. struktury

$status[u]$ - FRESH / OPEN / CLOSE

$d[u]$ - vzdálenost $s \rightarrow u$

$p[u]$ - ke kterému uzlu byl objeven

fronta - OPEN uzly

s - uzel od
kterého hledáme

Vlastnosti algoritmu

Časová složitost $O(|V|)$

výpočet hodnot nejkratších cest z počátečního
uzlu (pokud byl graf souvislý)

Prohledávání grafu do hloubky

DFS - Depth-First Search

uzly FRESH, OPEN, CLOSED

a mají časové značky s hodnotami $1..2 \cdot |V|$

$d[u]$ čas. n. otevření uzlu

$f[u]$ " uzavření " "

Složitost $O(|V| + |E|)$

Vlastnosti DFS

hrany se rozdělí na

- chromové - vedly do FRESH uzlu
- zpětné - vedly do OPEN uzlu
- dopředné - vedly ~~do~~ k OPEN u. do CLOSED uzlu $\rightarrow d[u] < d[w]$
- příčné - vedly do CLOSED uzlu $\rightarrow d[u] > d[w]$

v neorientovaném grafu pouze hrany

chromové a zpětné

výsledky DFS

Topologické uspořádání (hrany od uzlu s menším č. k uzlu s větším č.)

~~Algo~~
- průzkum kábelníků, při uzavírání uzlu tento uložíme na kábelník

po ukončení jsou na kábelníku uzly
v topologickém uspořád. - nemí být cykly
 $\rightarrow$ indikuje existence zpětných hran

Silné komponenty

- my vím se $f[u]$ pro u
vydvoří se G^- opacně orient
- DFS(G^-) od uzelů v pořadí klesajícím $f[u]$
- silné k.

(5)

Každý graf je zjednocením svých hran - kromě
grafu s diskrétními uzel

Problém 7 mostů - projít 7 mostů každý jednou
a vrátit se do počátku a



Hledáme pokrytí uzavřenými koly

Eulerův graf - každý uzel má sudý stupeň
 $\delta(u)$ sudé pro všechna $u \in V$

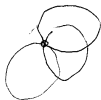
1: G je Eulerův graf $\Leftrightarrow G = \cup K_i$; $K_i \cap K_j = \emptyset$ (hranově)

2: $\Rightarrow$



$G - K_1$
skupné kolo uzel se sníží o 2
zůstane graf opět Eulerův
a tak až nic nezbyde

⇐



ka každou kružnici procházející
tímto bodem přibývá 2 kruhy,
s kterými incidují

V: graf lze pokrýt jedním neotevřeným obloukem
⇐ je souvislý a Eulerův

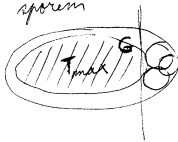
Ditto

⇒

✓ viz audio

⇐

sporem



G souvislý a Eulerův
předpoklad: nepodařilo
se pokrýt jedním obloukem

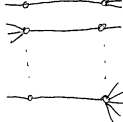
G - T_max

ve zbývajícím zbytku nějaké kružnice,
ořezy jede další připojit k oblouku
... musí jít pokrýt jedním obloukem

7 mostů v Královci podle zadání nemá řešení
- lze mosty pokrýt jedním otevřeným obloukem?

V: Necht' souvislý graf G má 2m uzelů lichého stupně. Pak lze jeho min. pokrytí tvořit m ošetřených sahů

D: m m ← by 2 stupiny uzelů l. stupně



~~každá~~ vždy dvojici propojit hranou

→ Eulerův graf

ke propojení navíc sahem

nově přidané hrany oddělily těmi původními

po jejich odstranění zbyde m kousků z toho sahů

- 7 mostů - 4 uzly lichého stupně
- minimální pokrytí 2 ošetřené sahy



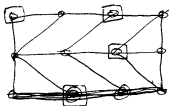
2 uzly lichého stupně

- pro OG

• Orientovaný Eulerův graf: $\delta^+(u) = \delta^-(u)$ pro každý uzel

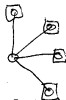
Nerávislá podmnož. uzlů

- mezi ~~lib~~ každými 2 uzly v této podmnož. není hrana



maximální nerávislá podmnož. uzlů

- nelze přidat další uzel, ~~bez~~ aniz by byla porušena nerávislost



jedna max. podm.

další max. podm.

V každém konečném grafu konečný počet max. podm.

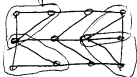
Nerávislost grafu

$$\alpha(G) = \max |I| \text{ pro } I \in \text{Ind}(G)$$

- nejvyšší mocnost k max. max. podm.

Klika grafu

každý ~~z~~ jeho maximální úplný podgraf



Kličky

Kritičnosť grafu $w(G)$

- počet vrcholů maximální kliky

platí $w(G) = \chi(-G)$

Dominující podmnožina



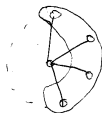
všechny slyšely ualy na
1 blak od D

$$D \cup \Gamma(D) = U$$

hledáme minimální podmnožiny

Dominance grafu

počet vrcholů minimální dominující podmnožiny



pro minimální dom.
podmnožiny

V: Nezávislá podmnož. $I \subseteq U$ je maximální $\Leftrightarrow$
podmnož. I je dominující.

Důsledek: $\beta(G) \leq \chi(G)$

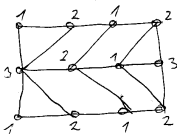
Příklady na využití

- šachovnice

- max počet dam, aby se neohrožovaly
- co nejméně dam, aby pokryly šachovnici
- když byl na vojně poručíkem ... (kolem 50 min)

Barvenost grafu

Barvit lze vršly nebo hrany
Ideálně barvené vršly nesmí být spojeny hranou



Dramatické číslo grafu G

$\chi(G)$

- minimální počet barev postačujících k obarvení

$$\forall: \chi(G) \cdot \alpha(G) \geq |V|$$

max. nezávisl. podm.

$$\chi(G) \geq \omega(G)$$

k. číslo větší než klitovost
klitka potřebuje n barev
(n počet jejích sousedů)

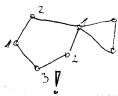
$$\chi(G) \leq \delta_{\max} + 1$$



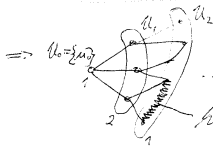
$\Delta_{\max}$ může být libovolně velké
 $\chi(G) \leq \Delta_{\max} + 1$ velmi přehnané

- $\chi(G) = 2 \Leftrightarrow G$ neobsahuje kružnici liché délky

D)



1. Liché délky
 $\rightarrow$ nelze 2 barvy



proto procházení
do třídy

hrana bránící obarvení
vzniknul by uzavřený
sled liché délky, který
obsahuje kružnici liché d.,
do ale nemáme - je to kóšer

Bipartitní graf - $\chi(G) = 2$
může se rozpadat do 2 tříd

Úplný bipartitní graf $K_{m,n}$
má všechny možné hrany

- G 10 uhlíků, $\chi(G) = 2$
 $? |H| \leq$

1. 9 $\leadsto$ 9
2. 8 $\leadsto$ 8
5. 5 $\leadsto$ 25

přecházel Debi Matyáš

⑥

Vzdálenost na grafu $d(u, v)$

delší nejkratší cesty mezi 2 uzel (počet hran)

$d(u, v)$ nejkratší celi číslo

$d(u, v) \geq 0$, $d(u, v) = 0$ když $u = v$

$d(u, v) = d(v, u)$

$d(u, v) \leq d(u, z) + d(z, v)$ (Δ nerovnost)

$d(u, v) > 1$, pak

Excentricita uzlu u v grafu G

$e(u, G) = \max d(u, v) \quad \forall v$

Poloměr grafu G

$r(G) = \min e(u, G) \quad \forall u$, střed(-y) grafu

Vzdálenost v OG

maňuje se nejkratší cesta - ale orientovaná

Grafy s (mnoh) ohodnocením $w: H \rightarrow \mathbb{R}^+$

w -delší orient. spojení $S = (h_1, h_2, \dots, h_n)$

$\sum w(h_i)$

$d_w(u, v)$ = w -delší nejkratšího orient. spojení

$+\infty$ - neexistuje spojení

$-\infty$ - prochází spojením se zápor. ohod.

Stromy a koky

V: Necht $G = \langle H, U \rangle$ je prostý graf. Potom

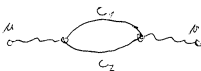
a) G je strom

b

c
kryt

D: a) $\Rightarrow$ b)

golem



2 různé cesty

$\rightarrow$ tvoří kružnici

$\rightarrow$ není strom

b) $\Rightarrow$ c)



mezi sebou jedinou cestu
mezi náby
 $\rightarrow$ není souvislý

c) $\Rightarrow$ d)

indukcí:

d) $\Rightarrow$ e)

101-1 hran potřebujem na souvislost -
nemůžeme plynout na kružnici

e) $\Rightarrow$ f)

přidáním libovolné hrany kružnice vznikne

f) $\Rightarrow$ a)

Kostra

strom, který obsahuje všechny vrchy
(faktor daného grafu, který je stromem)



kostra grafu má hrany na

- větve
- stěny - spojen s koster

Fundamentální soustava kružnic

kružnice, které vzniknou přidáním stěny do kostry

rekonvixtý graf - strom pro každou komponentu

- ještě navíc - LES

- Cyklické číslo (počet nekř. kružnic) $\Delta_{G-1} = 2$

$$\mu(G) = |H| - |V| + p \quad (p \text{ je počet komponent})$$

← CHYBA VE SLÁDKU

- Hodnost grafu (počet hran kostry nebo lesa grafu)

$$h(G) = |V| - p$$

Kořenový strom T_n (s koř. n) -

orient. strom, kde každá cesta $C(u, v)$ je
orientovanou cestou

($\forall$ koř. se dá dostat do lib. v po orient. cestě)

Kořenová křesť: křesť, která je kořenovým stromem.

Uspřádání stromu - následující každého uzelu...

Další přednáška

Pravidelné a binární stromy

Pravidelný strom stupně n ($n \geq 1$) - $S^+(n) = 0$ nebo n

každý

Úplný prav. strom stupně n hloubky k
všechny listy v hloubce k od kořene



V. Pro každé $n \in \mathbb{N}$ existuje prav. strom stupně 2
s n listy (má $n-1$ vnitř. uzlů a $2(n-1)$ hran)



Pro jiné stupně platit nemusí

(součet hloubek všech lidí)

vnější délka

$$E(T_M) = \sum h(v) \text{ přes listy } v$$

vnitřní délka

$$I(T_M) = \sum h(v) \text{ přes vnitřní uzly } v$$

V: Pro prav. strom splývá n a m vnitřními uzly platí

$$E = I \cdot (n-1) + n \cdot m$$

p. strom s nejmenší možnou délkou

vnitřní



vnější



Průchody lineárním stromem

- preorder: kořen, LS, PS

- inorder: LS, kořen, PS

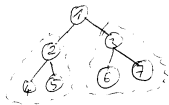
- postorder: LS, PS, kořen

1 2 4 5 3 6 7

4 2 5 1 6 3 7

4 5 2 6 7 3 1

listy vždy ve stejném
pořadí



Kódování binárních stromů

$((4\ 2\ 5)\ 1\ (6\ 5\ 7))$

(kód LS) 001 (kód PS)

Separabilita a planarita

Hranový řez - minimální $S \subseteq E$: $k(G-S) = k(G) - 1$
hrana se odstraní a vznikne více komponent

Artikulace grafu

$u \in V$: $G - \{u\}$ má více komponent než G

uzel, jehož odstraněním vznikne více komponent

V: množina hran incidentních s uzlem u v souvislém grafu je jeho hranovým řezem, právě když u není artikulace.

D: $k(G-u) = |U| - (P+1) = k(G-S)$ přesně tak

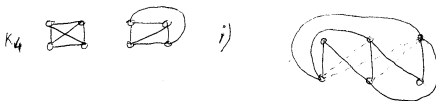
V: Hranový řez obsahuje alespoň jednu větev každé cesty obrátě sloup

neseperabilní graf

pro $\forall G, \exists G$ mají G a $G-G$, alespoň 2 uzly společné

Planární graf

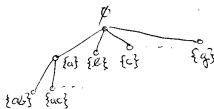
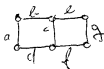
ke nakreslení v rovině bez křížení hran



(7)

minimální kostky a stromy

Hledání všech kostek



|U| - 1

kostky →

Strom všech možností, v každé větvi
karakterizuje, když daná podm. hran
obsahuje kružnici

efektivní test rovnosti kružnic

Binární rozklad - dynam. soubor podmnož. dané množ.

v každé podmnož. se nachází jeden reprezentant
přidávací třídy

- MAKE-SET(x)

- UNION(x, y)

- FIND-SET(x)

Power set $KOMP(G)$ - užití komponent NG

regulární rozdělení na jednod. uzel
pak bere hrany a pokud krajní uzel
regulární, ve stejné množině, tak se
jeho množina spojí do jedné
- pokud uzel hrany jsou ve stejné podmnožině,
jejím přidáním by vznikla kružnice

složitost $\Omega(|V| + |H|)$

- vyšší než procházení do hloubky / šířky
výhodné, pokud se graf mění
např. při přidání jedné hrany se provede
jen jednou vnitřní cyklus a je vyčíslen
množiny ~~rozděl~~ složeny do spjí seznamů,
každý prvek ještě reprezentanda své podm.
MAKE-SET / FIND-SET ... $O(1)$

UNION $O(\min(m, n))$ - pro každý seznam
udržoval odkaz i na poslední prvek
a počít prvků v něm

Agregovaná složitost m operací ($m \times \text{MAKE-SET}$)
 $O(m + n \cdot \lg m)$

složitost měnění hodnoty

$|V|$ -krát MAKE-SET

max $2|H|$ krát FIND-SET

$|V| - 1$ krát UNION

$$O(2|H| + 2|V| + |V| \cdot \lg |V|) = O(|H| +$$

možnost zmenšení složitosti

počítat heuristiky - v průměru lepší složitost

- shracování cesty při FIND-SET

- UNION podle hodnoty

Datové struktúry

$p[x]$ predchádzajúci uzel x
 $hod[x]$ hodnota (aprox. výšky)

prvky možno vložiť se sebou, pokiaľ
 koreň je reprezentantom
 predchádzajú koreňu je koreň



MAKE-SET

$p[x] \leftarrow x$
 $hod[x] \leftarrow 0$

UNION(x, y)

LINK (FIND-SET(x), FIND-SET(y))

LINK implementuje sa podľa reprezentantov
 podmnožín a menší hodnotu pripojím
 pod koreň s väčšou - prepisem toho
 koreň predchádzajú koreňu

Hodnotu sledujeme jen u reprezentanta
 - vždy vždy hod skopiuje: hod koreňu nového
 skopiuje se koreňu + 1

FIND-SET



pri prvom volaní nejake cesty
 ke koreňu se vracíme prosté
 vždy sa vrací priamo na koreň
 → skracovanie cesty

→ složitost $m \cdot \alpha(m \times \text{MAKE-SET}) : O(m \cdot \lg^* m)$ - skoro konštantná
 bolíak a log, aby menší nárast

Minimální kosť

- ohodnocení hran grafu - reálná čísla

$$w: H \rightarrow \mathbb{R}^+$$

minimální - součet ohodnocení hran je nejmenší
ze všech hran

jak najít?

- prohledat všechny kosťy :(
- ohodnotit kosť a ověřit minimálnost

V kosť T grafu G minimální, pro každou
řeznici t je $w(t) = \max_{h \in t} w(h)$ pro $h \in K$
(K je jediná hrana v $T \cup \{t\}$)

- každá řeznica musí mít aspoň jednu hranu kosť ležící
na hraně, vyřezané soudu řeznic
- řeznic je $|H| - |V| + 1$ - holbrát by se šest
musel provést - takže vysoká složitost

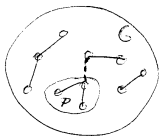
Praktické možnosti:

- odčítat hrany ???
- nějakou kosť a upravovat ???
- vyřezávat min kosť přidáváním hran !!!

Generický algoritmus $\text{GENERIC-MST}(G, w)$

jak hledat správnou hranu pro přidání?

Y. Necht P je podstrom...



vst ^{hlavní} část P a maximal
hrany, které vedou k P některým,
a těch vst nepřidat
a tu přidat k min kostce

Boincio - Kruskalův algoritmus

rovnostř. matematick Boincio řeší elektrifikaci
roslany i)

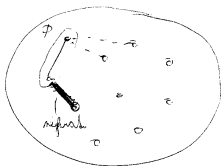
hrany seřadit do nekles. posloupnosti
postupně je brát a přijímat; jestli nevychází
kružnice

slož - časem $O(|H| \cdot |G| \cdot |H|)$ přiznání

$O(|H| \cdot |G| \cdot |V|)$: brzo

Jarníkovo - Primův algoritmus

- každou vybranou k jedné straně, postupně k ní přidává hrany



vezme nejkratší hranu
vycházející z již
včetně částí:

u každého uzlu - d - vzdál jeho chabete od P
p - předchůdce

čabetele vyhledávají prioritární frontu podle vzdál

- na začátku v ní všechny uzly, vzdál ∞ ,
je-li poč uzel ~~vzdál~~ vzdál 0 - vybere
se tedy jako první

(8)

Hledové algoritmy

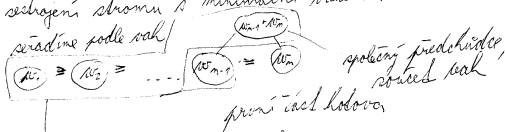
Hledový výběr - výběr podproblému a pak jeho řešení
optimální podstruktura

jiné použití - navrhování stromů

Huffmanův algoritmus

sestavení stromu s minimální váženou délkou

seřadíme podle vah



podproblém - problém vel. $n-1$

a tak se děje dokola

Aplikace - generování optimálního prefixového kódu

prefixový kód - žádné slovo není prefixem jiného slova

kódy kódovány různým počtem bitů podle četnosti výskytu

- vytvoří se optimální vážený strom

hrany dolova označ. 0, doprava 1

a kód znaku je ~~st~~ příslušná sekvence
úhel hran

Nejkratší cesty k jednoho uzlu

oklohovaný orientovaný graf

- nedostupné uzly - vzdál. ∞

- spojení se záporným cyklem - ~~ne~~ vzdál. $-\infty$

přičítání s nekonečnem

$$a + (-\infty) = -\infty \quad (a \neq \infty)$$

$$a + \infty = \infty \quad (a \neq -\infty)$$

$$\infty + (-\infty) = \infty$$

Varianty úlohy hledání nejkratších cest

$1 \rightarrow 1$ $1 \rightarrow n$ $n \rightarrow 1$ $n \rightarrow n$

Základní varianta algoritmu

inid Paths

Relaxace (u, v, w)



Dijkstra's algorithm

obecně použitelný pouze s kladnými ohodnoceními
v některých případech i pro hrany se záp. ohodnocením
skokový algoritmus prohlubování do šířky
do fronty mají se řadit podle vzdálenosti
s maximem neuvěřitelných málo
 $O(|H| \cdot \log |V|)$ základní varianta

- v. korespondenční diagram - hledání od kořene lze
použít i v případě záporně ohodnocených hran

Bellman-Ford's algorithm

formulován jako funkce s bool výsledkem,
při vrácení false nelze vzdálit více - je tam
záporně ohodnocený cyklus
true - záporný cyklus není, v daných
struktúrach složeny správné výsledky $d[u]$, $p[u]$
Relax konst složitost - nemáme prioritní frontu
složitost $O(|V| \cdot |H|)$

ličnost úpravy / krychlení

fronta - mely s úspěšným relax a bereme po
hrany & běhlo ušli

ukončení - upravení fronty

problém s řádk. cykly - mely by se celý pořádek,
muselo každý počítat, někdy byl
dozisk před $O(|V| \cdot |E|)$

jednoduché v acyklickém grafu
topologické seřazení

lineární dozisk v ká. na počtu ušli. a hran

nejkratší cesty mezi všemi páry ušli

repräsentace grafu pomocí upravené matice sousednosti
obsahuj. cesty hrany, nebo a - hrana není

Algoritmus Floyd - Marshall

malá matice vzdálenosti

panel - lze vyláchnout s jiné matice D

pro nalezení i cest - matice předchůdců

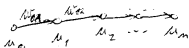
ke poznání i pro určování komponent grafu
 Reflexivně-transitivní uzavřen

Johnsonův algoritmus
 Dávké grafy se záporn. ohodnoc. hranami, časově lepší než Floyd-W
 kde $n \times$ Dijkstru, řeší $w(h) < 0$

Přehodnocení $w \rightarrow w'$ takové, že
 zachová minimálnost cest
 fce $h: V \rightarrow \mathbb{R}$ každému uzlu přiřadí nějakou hodnotu

$$\text{I: } w'(u, v) = w(u, v) + h(u) - h(v)$$

II



$$w'(L) = w_{01} + h(u_0) - h(u_1) + w_{12} + h(u_1) - h(u_2) + \dots$$

$\underbrace{\quad\quad\quad}_{=0}$

$$w'(L) = w(L) + h(u) - h(v)$$

určení fce h , aby všechny $w'(u, v)$ byly kladné
 upraveno G na G'

přidá se uzel s , ten do každého uzlu hrana
 váží 0, uzel s se vzdál s do každého uzlu;
 bylo vzdál, určitě fce h je vždy pro
 ten uzel

střednost $O(n \lg n)$ pro binární heap

9

Algebraické rovnosti

$\oplus$ - výběr z alternativ

$\odot$ - řádová cesta

$\otimes$ - navazování cest

1. triviální spojení (neobsahuj hranu)



Zobecněný F-W alg.

Toky v sítích

sít: $S = \langle G, q, s, \#d \rangle$

q .. kapacita hran

s .. zdroj sítě

d .. spotřebitel sítě

(zobrazení)

hok v síti S-choďnocení hraden hok, že

- hranou netíe více, než je její kapacita
- součet vstupních toků minimálního toku
je roven součtu toků vytekajících
(kromě zdroje a spotřebitele)

Velikost toku |f|

součet toků vytekajících ze zdroje minus
součet toků do zdroje vstávajících =
~~tok~~ odtěněn a spotřebitel

(10)

Minimální tok síti

saturované hrany dělí síť na 2 části

Algoritmus Ford-Fulkerson

Umělá inteligence

problém

popis stavu, se kterých se může vyjít
operace/akce - přechod mezi stavy
poč / konec stav

převáděno na graf

příslušný stavový prostor může být drowský
(neboe ukáží reprezentaci)

následující stavy generovány na základě aktuálního

hledání řešení

neinformované hledání - náhodné prohledávání

~~informované~~

- do šířky

- upor. vyhledem (uniform cost)

- do hloubky

- obousměrné - z poč. i konce.

stavu

graf známy jen o propátrané části, ostatní
nejdou známy, objevovány na základě operací

Prohledávání do šířky

↑ operátor, který pro lb. stav vygeneruje jeho následníky
slouží k - nebo obecně říci

při prohled. se dostane nejčastěji k uzlům
stejně vzdál jako cíl. stav
test - nový stav - slouží, nejlépe řečeno pomocí hash tabule

Prohled. do hloubky

výrazně méně paměťové nároky (lineární)
ALE - graf může být nekonečný, nemusel
by najít řešení

řešení - v každém kroku omezení hloubku,
při neúspěchu zvětšit a zkoušet znovu

Prohledávání upořádaným vyhledáváním (uniform cost)

předpokládá hodnotící funkci stavu $f(n)$,
podle těch do prioritní fronty
podobně Dijkstravi

je možné know-howování uzlů - dané hodnotící
fci

nalezení optim. řes - musí - hodnoty $f(n)$ neklesají
podle cesty řešení

Heuristické hledání

heuristická fee $h(n)$ - odhaduje zbývající vzdál. k cíli
označení

$g(n)$

$h(n)$

$f(n)$

přesné vzdál

$\hat{g}(n), \hat{h}(n), \hat{f}(n)$ aproximace

podle fee $f(n)$

$\hat{f}(n) = \hat{h}(n)$

Best/First - jde se ka nejlepšímu cestou,
vždy při správném odhadu nebespečné

$\hat{f}(n) = \hat{h}(n) + \hat{g}(n) \dots A^* (A\text{-star})$

uvazuje se

o projití csta, ~~střed~~

Průhledný alg. hledání - najde opt. řes (if exist.)

$h(n)$ nemí přecitovat vzdál - je konzistentní

varianty A^*

IDA* iterative deepening A^*
vždy omez. hl

SMA*

pro řešení
omez. paměti

Abstraktní heuristické hledání

- normálně se prohledávání směřuje k cíli/krátce
a ne k celu protýpého prohledávacího paprsku
- možno počítat heur. fci k malému protýpého
paprsku - čas. náročnost
- volí se referenční uzly, který se po několika
cyklech mění

Modely strojů, programů a výpočtů

(12) (13) jsem nebyl

Turingův stroj se vždy musí zastavit
(na přijatelném vstupu)

Polorozhodování - Lur. stroj se zastaví právě jen
pro slova daného jazyka

jazyk rekurzivně vyčíslitelný
použití, pokud nelze vyrobit stroj, který by se vždy zastavil

Třída složitosti

NP prob

nedeterministický polynomiální

vyřešení v exponenciálním čase, ale

že v polynomiálním čase ověřit, zda je něco řešení