

# Kapitola 1

## Grafy, stromy

[TODO nejaký hezký obrázek?] V této kapitole definujeme graf jakožto matematickou strukturu, popíšeme základní pojmy týkající se grafů a nastíníme možné vztahy mezi grafem a maticí. Dále definujeme strom, jakožto speciální případ grafu. Terminologie je převzata z [3]

### 1.1 Základní grafová terminologie

**Definice 1.** Mějme množinu  $V$  a množinu  $E = \{\{u, v\} \mid u, v \in V\}$ . Uspořádanou dvojici  $G := (V, E)$ , nazveme neorientovaný graf. Množinu  $V$  nazýváme množinou vrcholů grafu  $G$ , jejím prvkům říkáme vrcholy, množinu  $E$  nazýváme množinou hran grafu  $G$ , jejím prvkům říkáme hrany. Prvky hrany  $e$  označujeme jako vrcholy incidentní hraně  $e$  nebo koncové body hrany  $e$ . Říkáme, že hrana  $e = \{v, w\}$  spojuje vrcholy  $v$  a  $w$ .

Pokud neuvažujeme hrany jako nejvýše dvouprvkové množiny vrcholů, ale jako uspořádané dvojice  $(u, v)$ , nazýváme odpovídající graf **orientovaný**. Obvykle uvažujeme orientované a neorientované grafy zvlášť, ale je možné uvažovat i jejich kombinaci. Graf, v němž se vyskytují jak orientované tak neorientované hrany, nazýváme **smíšený**. Řekneme, že neorientovaný graf  $G$  je **úplný**, pokud  $\forall u, v \in V (\{u, v\} \in E)$ .

Povšimněme si, že v definici grafu není vyloučen případ, kdy jsou oba koncové body hrany shodné. Hrana je pak jednoprvkovou množinou a nazýváme ji **smyčkou** v grafu.

**Poznámka 1.** V neorientovaném grafu  $G = (V, E)$  platí, že jeho množina hran  $E$  je podmnožinou  $\binom{V}{2} \cup V$ , kde  $\binom{V}{2}$  značí množinu všech dvouprvkových podmnožin množiny  $V$ . V orientovaném grafu  $H = (W, F)$  je  $F$  podmnožinou množiny  $W \times W$ , tj. všech uspořádaných dvojic vrcholů z  $W$ .

**Stupněm vrcholu**  $v \in V$  rozumíme počet vrcholů spojených s vrcholem  $v$ , značíme  $d(v)$ . Množinu všech vrcholů, které jsou v grafu  $G$  spojeny s vrcholem  $v$  značíme  $\text{adj}_G(v)$ .

**Definice 2.** **Podgrafem** grafu  $G$  nazveme libovolný graf  $H = (W, F)$  který splňuje:  $W \subseteq V$ ,  $F \subseteq E$  a všechny vrcholy incidentní hranám z  $F$  náleží do  $W$ . Úplný podgraf grafu  $G$  nazýváme **klikou** v grafu  $G$ . Podgrafem grafu  $G$  **indukovaným** množinou vrcholů  $W$

nazveme takový podgraf  $G$ , který obsahuje všechny hrany grafu  $G$ , jejichž oba koncové body náležejí do  $W$ , značíme  $G(W)$ .

[TODO Potřebuju ohodnoceny, vázeny?]

**Definice 3.** Mějme graf  $G = (V, E)$  a zobrazení  $\omega : V \rightarrow \mathbb{R}$ , resp.  $c : E \rightarrow \mathbb{R}$ . Přidáním zobrazení  $\omega$ , resp.  $c$  ke grafu  $G$  dostaneme graf, který nazýváme **ohodnocený**, resp. **vážený** reálným ohodnocením.

**Definice 4** (TODO doplnit definici cesty bez cyklu!, cyklu, délka cesty).

Existuje-li mezi libovolnými dvěma vrcholy grafu cesta, řekneme, že graf je **souvislý**. **Vzdáleností** dvou vrcholů v souvislém grafu  $G = (V, E)$  nazveme minimální délku cesty mezi těmito dvěma vrcholy. **Vzdáleností** vrcholu  $v \in V$  od množiny vrcholů  $W \subset V$  nazveme minimální vzdálenost mezi vrcholem  $v$  a libovolným vrcholem náležícím do  $W$ .

[TODO potřebuju bipartitní?] [TODO potřebuju ctvercovou síť?] [TODO potřebuju faktorgraf?]

## 1.2 Strom

Nyní zavedme základní pojmy týkající speciální třídy grafů nazývané stromy [3]

**Definice 5. Stromem**  $T = (V, E)$  nazveme konečný souvislý neorientovaný graf bez cyklů. Pokud navíc v grafu  $T$  vyznačíme bod  $r \in V$ , nazýváme uspořádanou dvojici  $(T, r)$  **kořenovým stromem** a bod  $r$  nazveme kořenem tohoto stromu.

Z definice stromu je patrné, že každý vrchol  $v$  kořenového stromu  $(T, r)$  spojuje s kořenem tohoto stromu právě jedna cesta. Vrcholy ležící na této cestě nazveme **předchůdci** vrcholu  $v$ . Předchůdce vrcholu  $v$  různé od  $v$  nazýváme **vlastními předchůdci** vrcholu  $v$ . Vrcholy, jejichž předchůdcem je vrchol  $v$ , nazýváme **následníky** vrcholu  $v$ . Vrcholy bez následníků nazýváme **listy stromu**  $T$ , vrcholy alespoň s jedním následníkem nazýváme **vnitřní vrcholy** stromu.

**Definice 6. Podstromem** stromu  $T$  určeným vrcholem  $v$  nazveme indukovaný podgraf stromu  $T$  tvořený vrcholem  $v$  a všemi jeho následníky.

## 1.3 Vztah grafu a matice

Grafy a matice spolu úzce souvisí, což nám umožňuje převádět problémy na maticích na problémy na grafech a naopak. Nezanedbatelným praktickým důsledkem jejich vzájemného vztahu je i možnost používat grafové algoritmy při řešení některých maticových úloh, především může být tento přístup výhodný pro řídké matice. Dělení grafů může posloužit například při snaze o paralelizaci rozkladu matice.

Neorientovaný graf  $G = (V, E)$  s vrcholy  $V = v_1, \dots, v_m$  a hranami  $E = e_1, \dots, e_n$ . Tento graf lze reprezentovat pomocí matice dvěma základními způsoby. **Maticí sousednosti**, neboli adjacenční maticí, nazveme matici  $A_G$  o rozměrech  $m \times m$ , jejíž prvek na pozici  $(i, j)$  je definován jako:

$$(A_G)_{i,j} := \begin{cases} 1 & \text{existuje-li hrana spojující vrcholy } v_i, v_j \\ 0 & \text{jinak} \end{cases}$$

Maticí incidence grafu  $G$  nazveme matici o rozměrech  $m \times n$  definovanou následovně:

$$(\bar{A}_G)_{i,j} := \begin{cases} 1 & \text{je-li } v_i \text{ koncovým vrcholem hrany } e_j \\ 0 & \text{jinak} \end{cases}$$

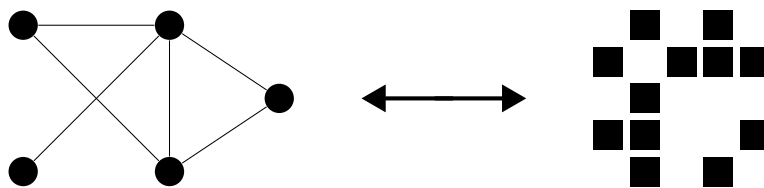
V kapitole rejspektral [TODO bude ref spektral?] budeme potřebovat Laplaceovu matici  $Q$  grafu  $G$ , která je definována následovně:

$$Q_{ij} := \begin{cases} -1 & \text{pro } i \neq j, (v_i, v_j) \in E \\ 0 & \text{pro } i \neq j, (v_i, v_j) \notin E \\ d(i) & \text{pro } i = j \end{cases}$$

Laplaceovu matici  $Q$  lze tedy vyjádřit jako  $Q = D - A_G$ , kde  $D$  značí diagonální matici se stupni jednotlivých vrcholů na diagonále.

Pokud chceme reprezentovat matici pomocí grafu, většinou nám stačí zachytit její strukturu. V takovém případě můžeme pro popis obecně nesymetrické matice  $A$  o rozměrech  $n \times n$  použít orientovaný graf s množinou vrcholů  $V = v_1, \dots, v_n$  a množinou hran  $E = \{(v_i, v_j) | a_{ij} \neq 0\}$ . V případě, že je matice  $A$  symetrická, můžeme ji analogickým způsobem reprezentovat pomocí neorientovaného grafu. Pokud bychom chtěli do grafu zanést i numerické hodnoty jednotlivých prvků matice, museli bychom použít ohodnocený graf.

[TODO rozhodnout, jestli tam tohle davat] Pro reprezentaci ne nutně čtvercové matice  $B$  o rozměrech  $m \times n$  můžeme také použít bipartitní graf  $G = (R, B, E)$  pro nějž platí  $|R| = m$ ,  $|B| = n$  a  $E = \{(i, j') \mid i \in R, j' \in B, a_{ij'} \neq 0\}$ . [end TODO]



Obrázek 1.1: Příklad grafu a jemu odpovídající struktury matice

chapterDělení grafů

V této kapitole z formálního hlediska popíšeme problém dělení grafu a definujeme pojmy s dělením spojené. Dále se budeme věnovat některým technikám používaným pro dělení grafů, konkrétně vylepšovacímu algoritmu podle Kernighana a Lina a algoritmu spektrálního dělení. Na závěr kapitoly zmíníme algoritmus metody vnořených řezů (Nested Dissection) jakožto konkrétní příklad víceúrovňového algoritmu pro dělení grafů a popíšeme i schéma víceúrovňového dělení grafu obecně.

Dělení grafů na dva nebo více podgrafů je praktický problém s bohatým teoretickým zázemím a mnoha aplikacemi. Může nám pomoci při řešení parciálních diferenciálních rovnic na moderních počítačových architekturách [8] a nezanedbatelnou roli hraje také při výrobě mikroprocesorů metodou VLSI nebo při řešení velkých systémů lineárních rovnic [2, 6].

Jako dělení grafu na  $k$  částí označujeme hledání rozkladu jeho množiny vrcholů na  $k$  podmnožin. Kritéria, která mají tyto podmnožiny splňovat, abychom toto rozdělení mohli označit jako optimální se mohou lišit. V této kapitole se zaměříme na klasickou definici dělení grafů, v dalších kapitolách poté bude popsán problém dělení grafů s jinými kritérii (například vzhledem k maticovým operacím) a s ohledem na více kritérií.

## 1.4 Formální definice dělení grafu

Standardní konvencí je omezit se při dělení grafu na jeho dělení na dva podgrafy, rozdělení na více částí lze poté dosáhnout rekurzivně. V neklasičtějších případech je problém dělení grafů na dva podgrafy definován následovně.

Mějme graf  $G = (V, E)$  s počtem vrcholů  $n$ . Hledání optimálního rozdělení grafu na  $k$  podgrafů je ve své podstatě hledání rozkladu  $V_1, V_2, \dots, V_k$  množiny vrcholů  $V$  splňujícího:

1.  $|V_i| = n/k$
2. Počet hran spojujících vrcholy ležící v různých podmnožinách je minimální možný.

Rozdělení grafu běžně zapisujeme jako vektor  $P$  o délce  $n$  takový, že pro každý vrchol  $v \in V$  je podgraf, v němž se vrchol  $v$  nachází, určen jeho  $v$ -tou složkou.

Ukazuje se, že rozhodovací problém pro optimální rozdělení grafu je NP-úplný [1]. Existují však algoritmy, které rozdělí graf v rozumném čase, přičemž kvalita jimi nalezeného rozdělení bude poměrně dobrá [4]. To ale neřeší problém samotného nalezení optimálního hranového nebo vrcholového separátoru [5]

## 1.5 Pojmy k dělení grafu

Mějme graf  $G = (V, E)$  a jeho rozdělení na dva podgrafy s množinami vrcholů  $A, B$ . Množinu hran, jejichž jeden koncový bod náleží do  $A$  a druhý do  $B$  nazveme hranovým separátorem, značíme  $\delta(A, B)$ . Vrcholovým separátorem grafu  $G$  nazveme podmnožinu množiny vrcholů  $V$  takovou, že odstraněním všech vrcholů náležících do této množiny z grafu  $G$  dojde k jeho rozdělení na dva podgrafy, které nejsou spojeny ani jednou hranou. Řešení problému optimální transformace mezi hranovým a vrcholovým separátorem, známe-li jeden z nich, můžeme nalézt v [7].

Pokud má graf  $G$  sudý počet vrcholů a množiny  $A$  a  $B$  mají stejný počet prvků, nazveme toto rozdělení bisekcí a velikost jeho hranového separátoru nazýváme šířkou bisekce.

## 1.6 Algoritmy pro dělení grafů

Algoritmy pro dělení grafů můžeme na základě jejich funkce rozdělit do dvou tříd.

První třídou dělicích algoritmů jsou algoritmy, které implementují přímo samotný proces rozdělení grafu na dva nebo více podgrafů. Do této třídy můžeme zařadit například algoritmy, které pro svou funkci potřebují souřadnice vrcholů grafu v rovině či prostoru (např. Inerciální dělení) [9]. Pro nás však bude zajímavé především spektrální dělení.

Druhou třídou jsou tzv. vylepšovací algoritmy, které neřeší přímo problém samotného rozdělení grafu. Jejich úkolem je vylepšit již existující rozdělení grafu tak, aby výsledné rozdělení bylo blíže optimálnímu. Základním vylepšovacím algoritmem je algoritmus podle Kernighana a Lina.

Algoritmy z těchto dvou tříd lze kombinovat. Nejprve necháme algoritmus z první třídy najít rozdělení grafu a poté vylepšovacím algoritmem zkusíme toto rozdělení optimalizovat.

Při dělení velkých grafů je často využíváno víceúrovňové schéma, skládající se ze tří fází. V první fázi je zredukována velikost grafu, ve druhé fázi je provedeno samotné dělení grafu pomocí některého z výše zmíněných algoritmů a ve třetí fázi dochází k promítnutí nalezeného rozdělení na původní graf.

### 1.6.1 Algoritmus podle Kernighana a Lina

Algoritmus podle Kernighana a Lina (KL algoritmus) vznikl v roce 1970 s cílem dělit elektrické obvody na kartách. Jak již bylo zmíněno, jedná se o vylepšovací algoritmus, pro jeho funkci je tedy nutné poskytnout mu vstupní rozdělení grafu. Toto rozdělení lze zvolit libovolně, ale v praxi je výhodné, aby představovalo rozumnou aproximaci optimálního rozdělení. Klasicky je KL algoritmus nebo jeho modifikace používán pro vylepšení rozdělení nalezeného jiným grafovým algoritmem. Nezanedbatelnou úlohu hraje také při víceúrovňovém dělení grafů, viz. ???. V této práci je algoritmus podle Kernighana a Lina zmíněn proto, že jeho modifikace mohou být využity pro vylepšování rozdělení nejen s ohledem na váhu jednotlivých hran, ale i s ohledem na jiné aspekty, které budou specifikovány níže.

Mějme graf  $G = (V, E)$  a rozklad jeho množiny vrcholů na podmnožiny  $A$  a  $B$ . Základem KL algoritmu je cyklus, v jehož každé iteraci dojde k výměně určitého počtu vrcholů z množiny  $A$  za stejný počet vrcholů z množiny  $B$  tak, aby byla snížena velikost hranového separátoru.

Vrcholy vhodné pro výměnu vybíráme na základě zisku spojeného s přesunem vrcholu  $v$  z množiny  $A$  do množiny  $B$ . Tento zisk definujeme jako rozdíl počtu hran (případně vah hran) spojujících vrchol  $v$  s vrcholy v množině  $B$  a počtu hran spojujících vrchol  $v$  s vrcholy v množině  $A$ :

$$D(v) := \sum_{\substack{(u,v) \in E \\ P[u] \neq P[v]}} w(u,v) - \sum_{\substack{(u,v) \in E \\ P[u] = P[v]}} w(u,v).$$

Pokud přesuneme vrchol s kladným ziskem, dojde ke zmenšení hranového separátoru.

Abychom mohli popsat reálný zisk při výměně vrcholů, je vhodné zavést si ještě pro  $u, v \in V$  hodnotu  $C_{uv}$ .

$$C_{uv} = \begin{cases} 1 & u, v \in E \\ 0 & \text{jinak} \end{cases}$$

Pak reálný zisk při výměně vrcholů  $u$  a  $v$  ležících ve vzájemně různých částech rozdělení grafu je zřejmě

$$g_{uv} = D(u) + D(v) - 2C_{uv}.$$

Základní varianta KL algoritmu se skládá ze dvou do sebe vnořených cyklů, z nichž vnější běží do té doby, dokud se rozdělení grafu zlepšuje a obsahuje tyto kroky

1. Vypočítáme zisk jednotlivých vrcholů.
2. Postupně spárujeme všechny vrcholy z  $a_i \in A$  s vrcholy s  $b_i \in B$  tak, aby reálný zisk  $g_i$  při výměně právě spárované dvojice vrcholů za předpokladu, že by všechny dvojice vrcholů nalezené před ní byly vyměněné, byl maximální.
3. Nalezneme takové  $k \in 1, \dots, \min |A|, |B|$ , aby  $\sum_{j=1}^k g_i$  byla maximální.
4. Vyměníme vrcholy  $a_1, \dots, a_k$  s vrcholy  $b_1, \dots, b_k$ .

Hlavní nevýhodou KL algoritmu je, že pro svou funkci potřebuje počáteční rozdělení grafu. Jeho výsledky pro různá počáteční rozdělení se mohou lišit. Kvůli tomu se algoritmus podle Kernighana a Lina obvykle používá v kombinaci s jiným algoritmem pro dělení grafů, případně jako součást většího celku (např. víceúrovňových algoritmů, viz ??).

# Kapitola 2

## Různé

### 2.1 Číslování

#### 2.1.1 Číslování vrcholů grafu v závislosti na vzdálenosti od separátoru

V této podkapitole popíšeme nejjednodušší metodu číslování vrcholů podgrafu, který vznikl rozdělením původního grafu na  $n$  částí. Tuto metodu lze používat samostatně, ale vzhledem k její povaze ji lze využít i pro vylepšení ostatních metod očíslování grafu, například ji lze kombinovat s metodou minimálního stupně.

Mějme graf  $G = (V, E)$  a jeho vrcholový separátor [TODO znaceni], jehož odebráním se graf rozpadne na  $n$  podgrafů  $G_1, \dots, G_n$ . Popíšme číslování vrcholů podgrafu  $G_i$ :

1. Položme  $j := 1$ .
2. Nalezneme neočíslovaný vrchol  $v$  grafu  $G_i$  takový, že jeho vzdálenost od vrcholového separátoru v grafu  $G$  je maximální.
3. Tomuto vrcholu dáme číslo  $j$ , položíme  $j := j + 1$ .
4. Pokud jsou všechny vrcholy očíslovány, skončíme, jinak se vrátíme na krok 2

Z algoritmu je vidět, že výsledné očíslování vrcholů grafu nemusí být jednoznačné, protože pokud nalezneme dva nebo více vrcholů, jejichž vzdálenost od separátoru je shodná, můžeme je očíslovat v libovolném pořadí.

#### 2.1.2 Číslování vrcholů pomocí metody minimálního stupně

Metoda minimálního stupně je jednoduchým algoritmem pro nalezení očíslování grafu. Algoritmus pro hledání očíslování grafu pomocí této metody je následující:

1. Mějme graf  $G = (V, E)$  a položme  $j := 1$ .
2. Nalezneme neočíslovaný vrchol  $v$  grafu  $G$  s nejmenším stupněm a přiřadíme mu číslo  $j$ .

3. Přidáme hrany mezi vrcholy z  $\text{adj}_G(v)$  tak, aby  $\text{adj}_G(v)$  byla klika v grafu  $G$ .
4. Pokud nejsou všechny vrcholy očíslované, zvětšíme  $j$  o 1 a vrátíme se na krok 2.

Očíslování vrcholů grafu  $G$  pomocí tohoto algoritmu není jednoznačné, protože vrcholů s minimálním stupněm může být více.

Pokud máme rozdělení  $G_1, \dots, G_n$  grafu  $G$  s vrcholovým separátorem [TODO značení], můžeme pro očíslování části  $G_i$  použít číslování vrcholů pomocí metody minimálního stupně, kde při výběru vrcholu ve 2. kroku přidáme kritérium vzdálenosti od separátoru popsané v 2.1.1. Nejprve tedy nalezneme množinu všech vrcholů grafu  $G$ , které mají minimální stupeň a poté mezi nimi zvolíme ten, který má nejmenší stupeň.

### 2.1.3 Topologické číslování vrcholů stromu

**Definice 7.** Mějme graf  $G = (V, E)$ , který je stromem. Očíslování jeho vrcholů nazveme topologickým právě tehdy, když pro každý vrchol  $v \in V$  platí, že libovolný následník vrcholu  $v$  ve stromu  $G$  má nižší číslo než vrchol  $v$ .



## Kapitola 3

# Eliminační stromy

V této kapitole se budeme zabývat eliminačními stromy a jejich významem pro rozklady řídkých matic. Eliminační stromy při rozkladu matic hrají důležitou roli, protože nám dávají informaci o zaplnění v Choleského faktoru matice bez toho, abychom museli počítat jednotlivé numerické hodnoty. Lze tedy díky nim jednoduše porovnávat vhodnost zvoleného uspořádání řádků a sloupců matice pro Choleského rozklad.

V této kapitole bez újmy na obecnosti předpokládáme, že matice, jejíž Choleského rozklad chceme napočítávat, je ireducibilní, a tedy přidružený graf této matice je souvislý.

### 3.1 Definice eliminačního stromu matice

Nejprve se omezme na ireducibilní, pozitivně definitní, symetrickou matici  $A_T$  o rozměrech  $n \times n$ , jejíž přidružený graf  $G(A_T)$  je strom. V tomto případě je  $A_T$  tzv. perfektní eliminační matice, tj. existuje permutační matice  $P$  taková, že Choleského rozklad matice  $PA_TP^T$  nebude obsahovat žádné zaplnění [10] (Matici  $PA_TP^T$  můžeme vnímat pouze jako přechíslování řádků a sloupců matice  $A_T$ ). Aby při choleského rozkladu matice  $A_T$  nedošlo k žádnému zaplnění, stačí když pomocí topologického číslování očíslovujeme vrcholy jí přidruženého grafu (z předpokladu se jedná o strom) a řádky a sloupce matice  $A_T$  seřadíme odpovídajícím způsobem. Pak zjevně platí, že matice  $A_T$  má, s výjimkou posledního řádku, pod diagonálou vždy právě jeden nenulový prvek. Díky tomu můžeme definovat pro matici  $A_T$  funkci  $\text{PARENT} : \{1, \dots, n\} \rightarrow 1, \dots, n$  následovně:

$$\begin{aligned} \forall j \in \{1, \dots, n-1\} \quad \text{PARENT}[j] &:= p \quad \Leftrightarrow \quad a_{p,j} \neq 0 \wedge p > j \\ \text{a speciálně: } \text{PARENT}[n] &:= 0. \end{aligned}$$

Zřejmě ve stromu přidruženém k matici  $A_T$  platí, že předchůdcem vrcholu  $x_j$  je vrchol  $x_{\text{PARENT}[j]}$ .

Většinou však nepracujeme s maticemi, jejichž přidružený graf by byl stromem. Zavedeme tedy konstrukci pro libovolnou řídkou, ireducibilní, pozitivně definitní, symetrickou matici  $A$  o rozměrech  $n \times n$ . Předpokládejme, že známe Choleského rozklad této matice, tj.  $A = LL^T$ . Maticí se zaplněním nazveme matici  $F$  definovanou jako  $F = L + L^T$ . Dále

zavedeme matice  $L_t$  a  $F_t$  následovně.  $L_t$  je matice vzniklá z  $L$  tím, že v každém sloupci vynulujeme všechny prvky pod diagonálou kromě prvku s nejnižším řádkovým indexem a  $F_t = L_t L_t^T$ .

Z definice  $F_t$  vidíme, že se jedná o matici, jejíž přidružený graf  $G(F_t)$  je strom.

**Definice 8.** Eliminačním stromem matice  $A$  nazveme graf  $G(F_t)$  popsany výše, značíme  $T(A)$ . Podstrom  $T(A)$  s kořenem  $x_j$  značíme  $T[x_j]$ . Množinu vrcholů tohoto stromu značíme také  $T[x_j]$ .

Díky této definici můžeme definici funkce PARENT přirozeně rozšířit na matici  $A$  následovně:

$$\text{PARENT}[j] := \min\{i > j \mid l_{i,j} \neq 0\},$$

kde  $l_{i,j}$  označuje  $i, j$ -tý prvek matice  $L$ .

**Pozorování 1.** Přímo z definice plyne, že  $T(A)$  a  $T(F)$  jsou identické.

**Pozorování 2.** Pokud  $x_i$  je vlastním předchůdcem  $x_j$  v eliminačním stromu, pak  $i > j$ .

**Tvrzení 1.** Pro  $i > j$  závisí numerické hodnoty sloupce  $L_{\bullet i}$  na sloupci  $L_{\bullet j}$  právě tehdy, když  $l_{i,j} \neq 0$ .

*Důkaz.* Tvrzení plyne přímo ze sloupcového algoritmu [???

□

# Literatura

- [1] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. Freeman & Co., 1979.
- [2] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307, 1970.
- [3] A. Koubková and V. Koubek. *Datové struktury 1*. Praha: Matfyzpress, 1 edition, 2011.
- [4] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math.*, 36(2):177–189, 1979.
- [5] J. W. H. Liu. A graph partitioning algorithm by node separators. *ACM Trans. Math. Software*, 15(3):198–219, 1989.
- [6] S. Pissanetzky. *Sparse matrix technology*. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], London, 1984.
- [7] A. Pothen and Ch.-J. Fan. Computing the block triangular form of a sparse matrix. *ACM Trans. Math. Software*, 16(4):303–324, 1990.
- [8] A. Pothen, H. D. Simon, and K. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11(3):430–452, 1990.
- [9] Alex Pothen. Graph partitioning algorithms with applications to scientific computing. In *Parallel numerical algorithms (Hampton, VA, 1994)*, volume 4, pages 323–368. Kluwer Acad. Publ., Dordrecht, 1997.
- [10] D. J. Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. pages 183–217, 1972.