Diskrétní matematika

Petr Kovář
petr.kovar@vsb.cz

Vysoká škola báňská – Technická univerzita Ostrava

zimní semestr 2022/2023
DiM 470-2301/01, 470-2301/03*, 470-2301/05

O tomto souboru

Tento soubor je zamýšlen především jako pomůcka pro přednášejícího. Řadu důležitých informací v souboru nenajdete, protože přednášející je říká, ukazuje, případně maluje na tabuli. Přednášky jsou na webu k dispozici, aby studenti mohli snadno dohledat probíraná témata z přednášek, které zameškali.

Pro samostatné studium doporučuji skripta:

- M. Kubesa: Základy diskrétní matematiky, výukový text
- P. Kovář: Úvod do teorie grafů, výukový text

Pro přípravu ke zkoušce a písemkám doporučuji cvičebnici:

- P. Kovář: Cvičení z diskrétní matematiky, sbírka příkladů

Vše na http://homel.vsb.cz/~kov16/predmety_dm.php

Přehled přednášky

Kapitola Stromy

- motivace
- základní vlastnosti stromů
- kořenové stromy
- isomorfizmus stromů
- kostry grafů

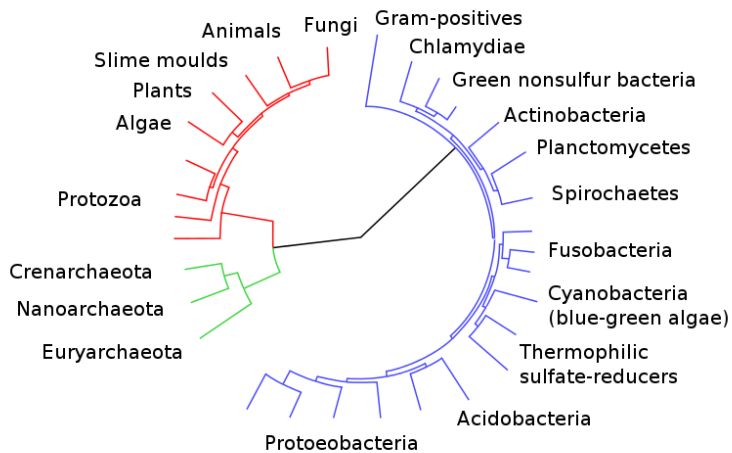
Kapitola Stromy

Motivace

Jedním z nejběžnějších útvarů v přírodě i v matematice jsou stromy (objekty se „stromovou“ strukturou).

Existuje celá řada motivací, které můžeme popsat „stromem“.

- rodokmeny
- evoluční strom
- elektrické rozvody
- hierarchické struktury (šéf a podřízený)
- větvení při vyhledávání



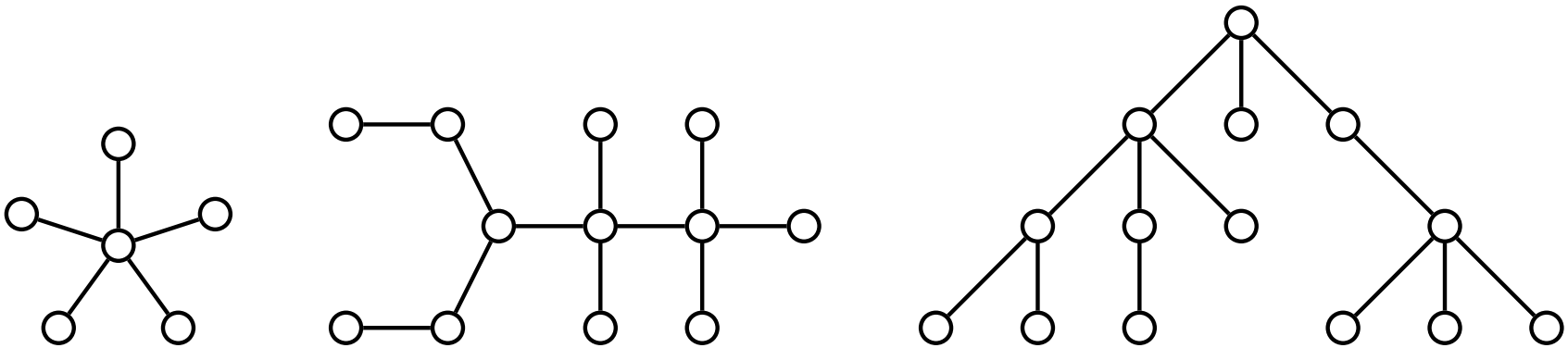
Společný rys: neobsahují „cyklus“.

Základní vlastnosti stromů

Řekneme, že graf acyklický, jestliže neobsahuje cyklus (kružnici) jako podgraf, tj. jestliže žádný jeho podgraf není isomorfní s cyklem.

Definice

Souvislý acyklický graf je nazývá strom.
Les je graf, jehož komponenty jsou stromy.



Poznámka k názvosloví:

- Les je (jednoduchý) acyklický graf.
- Strom je souvislý les.

Což nezní tak pěkně. . .

Vrcholům stupně 1 budeme říkat list.
Ostatním vrcholům budeme říkat nelistové vrcholy.

Lemma

Každý strom s více než jedním vrcholem má alespoň jeden list.

Důkaz Souvislý graf s více než jedním vrcholem nemůže mít vrchol stupně 0. Vezmeme strom T a v něm libovolný vrchol v. Sestrojíme co nejdelší tah S v T, který začíná ve v. S začneme libovolnou hranou vycházející z v (jistě existuje, proč?). V každém dalším vrcholu u tahu S

- buď u má stupeň alespoň 2 a tah S prodloužíme o další hranu a postup opakujeme,
  (všimněte si: v tahu S se nemůže zopakovat vrchol, protože v T neexistuje cyklus)
- nebo v u je stupně 1 (tah končí hledaným listem).

Protože T je konečný, jistě takový list ve stromu T najdeme. □

Poznámka

Není těžké ukázat, že v každém netriviálním stromu existují alespoň dva listy.

Otázky

- Jak se jmenuje strom, který má právě 2 vrcholy stupně 2 a žádný vrchol stupně vyššího?
- Existuje strom, který má vrchol stupně k a méně než k vrcholů stupně 1?
- Umíte vaše tvrzení z předchozího bodu dokázat?
- Kolik musíme vynechat hran z grafu K_n, abychom dostali strom?

Věta

Strom s n vrcholy má právě $n-1$ hran.

Důkaz

Postupujeme indukcí vzhledem k počtu vrcholů n.

Základ indukce: (Triviální) strom s jedním vrcholem má $n-1=0$ hran.

Indukční krok: Mějme netriviální strom T s $n>1$ vrcholy.
Předpokládejme, že každý strom s méně než n vrcholy má o jednu hranu méně než vrcholů.
Strom T má alespoň jeden list v (Lemma), označme $T'=T-v$ graf, který vznikne z T odebráním vrcholu v (tzv. „oholením vrcholu“).

- Odebráním listu neporušíme souvislost grafu (žádná cesta mezi dvěma vrcholy různými od v nevede vrchol stupně 1), T' je souvislý.
- Odebráním vrcholu/hrany nevznikne cyklus, T' je také acyklický.

To znamená, že T' je strom s $n-1$ vrcholy a podle indukčního předpokladu má o jednu hranu méně než vrcholů. Strom T' má tedy $(n-1)-1$ hran a původní strom T má o jeden vrchol a jednu hranu více, tj. strom T má $(n-1)-1+1=n-1$ hran.
Podle principu matematické indukce je tvrzení dokázáno. □

Příklad

V databázi je 12 záznamů (objektů) a 34 vazeb mezi nimi. Chtěli bychom strukturu databáze přehledně znázornit grafem, ve kterém jsou objekty znázorněny jako vrcholy a vazby jako hrany.
a) Bude takový graf stromem?
b) Můžeme tvrdit, že takový výsledný graf bude souvislý?

a) Výsledný graf jistě nebude stromem, ale musí obsahovat cykly. Strom s 12 vrcholy by musel mít právě 11 hran (vazeb).
Ani kdyby vazeb bylo 11, nemohli bychom tvrdit, že výsledný graf bude strom, proč?

b) Výsledný graf může, ale nemusí být souvislý. Výsledek velmi závisí na struktuře dat.
Mohlo by se stát: graf bude mít jednu komponentu s 9 vrcholy a 3 izolované vrcholy (K_9 má $\binom{9}{2} = 36$ hran, můžeme vynechat 2 hrany).

Pokud by daný graf měl 12 vrcholů a více než 55 hran, musí už být souvislý. K_{12} má 66 hran a je hranově 11-souvislý. Vynecháním libovolných méně než $66 - 55 = 11$ hran zůstane výsledný graf souvislý.

O důkazu vět ve tvaru implikace $A \Rightarrow B$

A je předpoklad věty, B je tvrzení věty.

Důkaz přímý spočívá v řadě platných implikací.

$$A = A_0 \Rightarrow A_1 \Rightarrow A_2 \Rightarrow \cdots \Rightarrow A_n = B$$

Důkaz nepřímý spočívá v přímém důkazy věty $\neg B \Rightarrow \neg A$, který má stejnou tabulku pravdivostních hodnot jako původní výrok $A \Rightarrow B$.

$$\neg B = A_0 \Rightarrow A_1 \Rightarrow A_2 \Rightarrow \cdots \Rightarrow A_n = \neg A$$

Důkaz sporem je na první pohled připomíná nepřímý důkaz, neboť pracujeme s negací tvrzení $\neg B$. Při pečlivější úvaze vidíme rozdíl: předpokládáme, že platí současně předpoklad A a negace tvrzení $\neg B$. Řadou platných implikací dojdeme k tzv. sporu. Sporem rozumíme situaci, kdy by měl platit výrok V a současně také jeho negace $\neg V$, což není možné.

$$A \wedge \neg B \Rightarrow \cdots \Rightarrow V \wedge \neg V$$

Ke sporu vedla negace tvrzení $\neg B$, proto platí tvrzení B.

Věta

Mezi každými dvěma vrcholy stromu existuje právě jedna cesta.

Důkaz Postupujeme sporem. Budeme vycházet z platnosti předpokladu věty (T je strom) a negace tvrzení (mezi některými vrcholy stromu T neexistuje žádná nebo existuje více než jedna cesta). Dostaneme spor.

Stromu je T souvislý graf, proto víme, že mezi každými dvěma vrcholy u, v stromu T vede nějaká cesta.
Pro spor předpokládáme, že mezi některými vrcholy u, v existuje více takových cest. Sjednocení těchto cest je uzavřený sled ve stromu T, ze kterého (vynecháním případných opakujících se hran a vrcholů) dostaneme cyklus, který je podgrafem ve stromu T.
Dostáváme spor, neboť T je strom, který neobsahuje cyklus.

Negace tvrzení vede ke sporu. Proto mezi vrcholy u a v existuje ve stromu právě jedna cesta. □

Poznámka

Cesty z u do v a "opačnou" cestu z v do u považujeme za jedinou cestu mezi u, v.

Už víme, že strom s n vrcholy obsahuje $n - 1$ hran. Přidáním jedné hrany

- neporušíme souvislost,
- ztratíme acykličnost.

Strom s přidanou hranou musí obsahovat cyklus, ukážeme že takový cyklus bude jediný.

Důsledek

Přidáním jedné (nové) hrany do stromu (s alespoň třemi vrcholy) vznikne graf s právě jedním cyklem.

Důkaz Mějme strom T a nějaké dva jeho vrcholy u, v, mezi kterými není hrana.

Přidáním hrany uv vznikne právě jeden cyklus spojením uv a jediné cesty mezi vrcholy u, v ve stromu T (jediné podle předchozí věty). □

Poznámka

Přidáme-li do stromu alespoň dvě hrany, závisí počet vzniklých cyklů na tom, kam hrany přidáme.

Máme dán nějaký souvislý graf G. Při určování hranové souvislosti jsme se ptali, kolik nejméně stačí z grafu odstranit hran, aby se G stal nesouvislý. Má smysl se ptát,

- kolik nejvíce hran můžeme z grafu G odstranit, aby G zůstal souvislý, nebo naopak
- kolik nejméně hran musí v grafu G zůstat, aby byl souvislý.

Právě stromy jsou grafy, které jsou souvislé a už z nich nemůžeme odebrat žádnou hranu, aniž by se porušila souvislost.

Věta

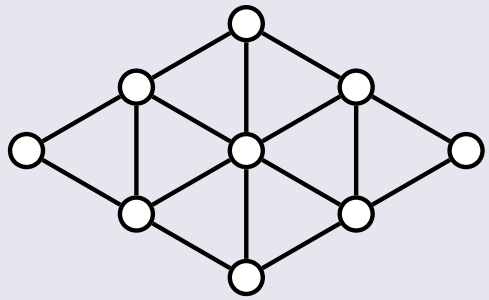
Strom je *minimální souvislý* graf (s daným počtem vrcholů).

Důkaz Strom je souvislý podle definice. Pokud souvislý graf obsahuje cyklus, zůstane souvislý i po vynechání některé hrany cyklu. Proto je minimální souvislý graf acyklický a je tedy stromem.

Naopak, pokud by vynecháním hrany uv ze stromu T vznikl souvislý graf, pak by mezi vrcholy u, v v T existovaly dvě cesty: cesta z u do v v $T \setminus uv$ a hrana uv. To by byl spor s předchozí větou.
Proto je strom minimálním souvislým grafem na daných vrcholech. □

Příklad

Kolik nejvíce hran můžeme odstranit z grafu G na obrázku, aby výsledný graf zůstal souvislý?



Graf G se stupni $(6, 4, 4, 4, 4, 3, 3, 2, 2)$.

Podle předchozí věty víme, že výsledný graf po odebírání hran bude stromem.
Protože graf G má 9 vrcholů a 16 hran, tak podle věty o počtu hran stromu víme, že můžeme odstranit nejvíce 8 hran.

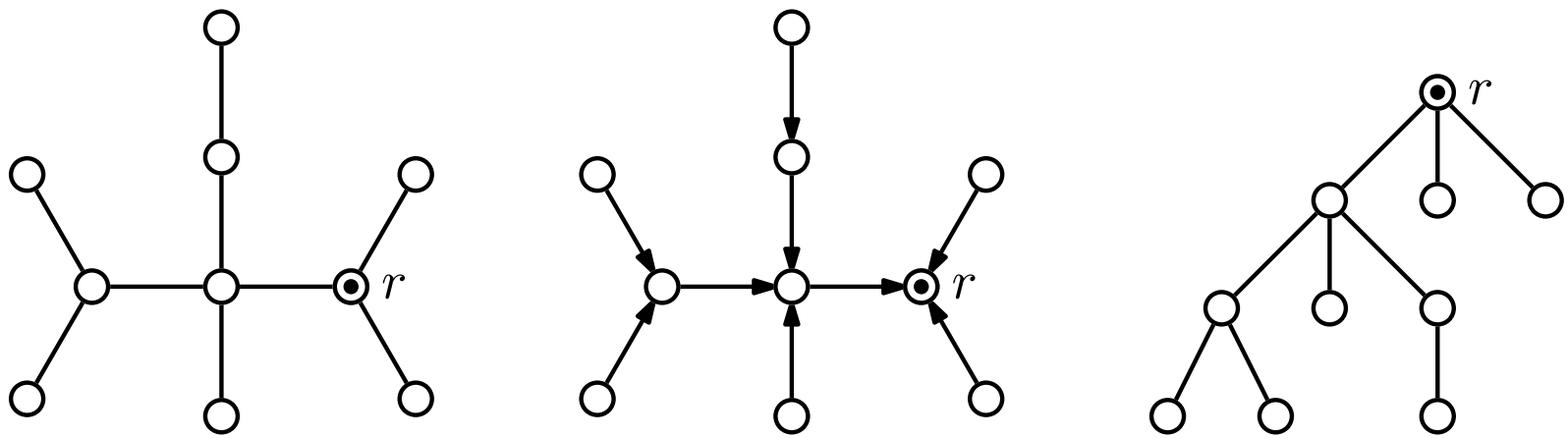
Z grafu na obrázku můžeme dokonce odstranit takových 8 hran, že odstraněné hrany tvoří spolu s vrcholy grafu G souvislý faktor a současně ponechané hrany tvoří souvislý faktor. Najdete takových 8 hran?

Kořenové stromy

Někdy je užitečné při práci se strukturou typu „strom“ označit význačný vrchol, tzv. kořen, („začátek“ uložených dat). Kořenové stromy mají motivaci i v rodokmenech, odtud vychází i terminologie.

Definice

Kořenovým stromem nazveme strom T spolu s vyznačeným kořenem $r \in V(T)$. Značíme (T, r), říkáme „strom T s kořenem r“.

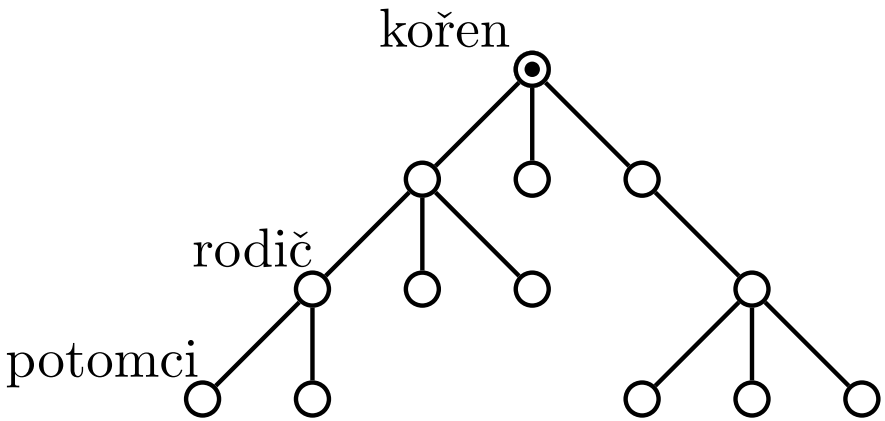


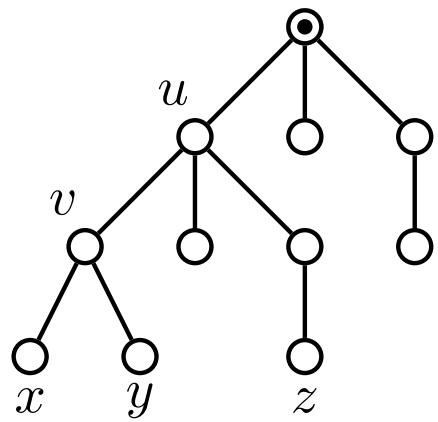
Nezaměňovat „strom“ a „kořenový strom“, který obsahuje informaci navíc.

Kořen budeme kreslit nejvýše.

Definice

Mějme dán kořenový strom (T, r) a v něm dvojici vrcholů u, v, kde vrchol u je sousední s vrcholem v na cestě z vrcholu v ke kořenu r. Pak u nazýváme rodičem vrcholu v a v nazýváme potomkem vrcholu u.





Někdy použijeme další přirozené názvy „prarodič“, „sourozenec“, . . .

Všimněte si: volbou kořene se může vzájemný vztah vrcholů změnit.

Definice

V netriviálních kořenových stromech se vrcholy bez potomků nazývají koncové vrcholy.

Všimněte si: koncové vrcholy jsou listy, ale ne každý list je koncovým vrcholem.

Definice

Centrem stromu T rozumíme vrchol nebo hranu v daném stromu T, které určíme následujícím postupem:

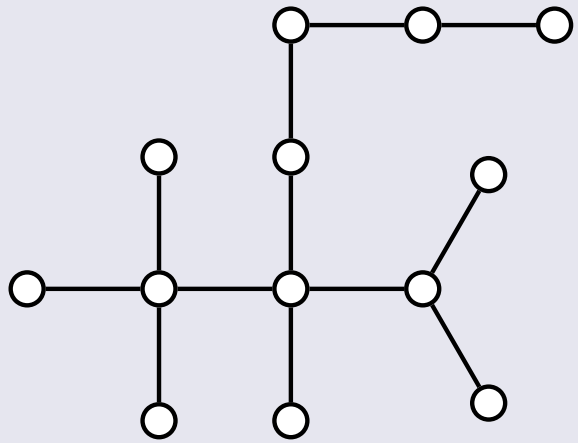
1. Jestliže má strom T jediný vrchol v, je v centrum stromu T. Pokud má strom T dva vrcholy, je centrem stromu T hrana spojující tyto dva vrcholy. Jinak přejdeme k bodu 2.
2. Vytvoříme (menší) strom $T' \subset T$ oholením všech listů stromu T a vracíme se na předchozí bod 1.

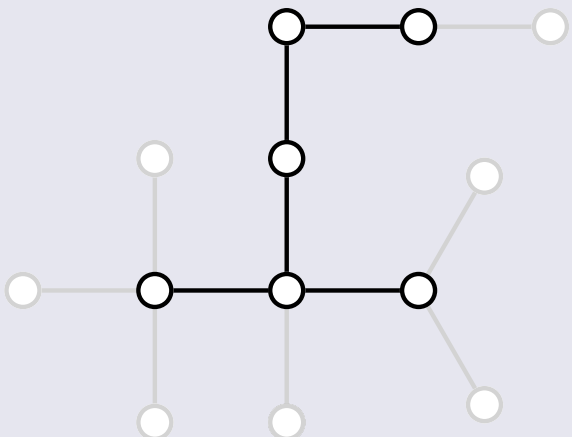
Rekurzí získané centrum stromu T' bude zároveň centrem stromu T.

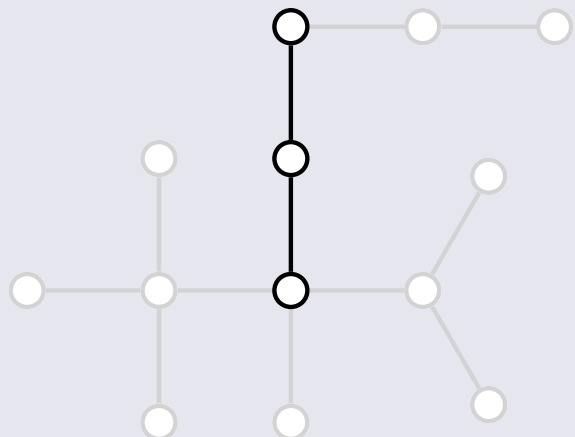
Procesu, kdy ve stromu označíme všechny jeho listy a potom je odstraníme (i s příslušnými hranami), se říká oholení stromu.

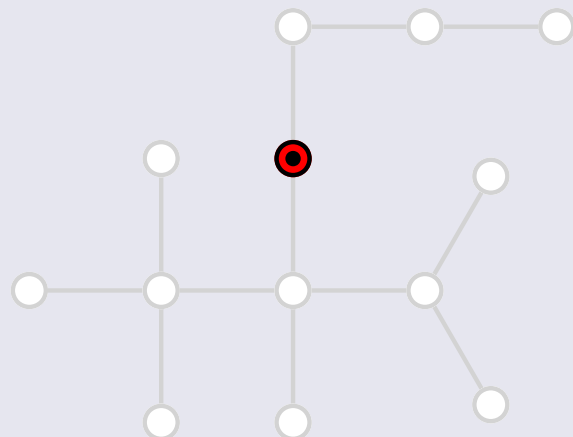
Příklad

Určete centrum stromu T_1.



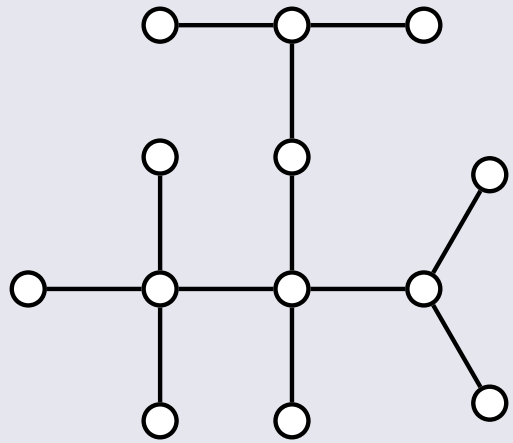


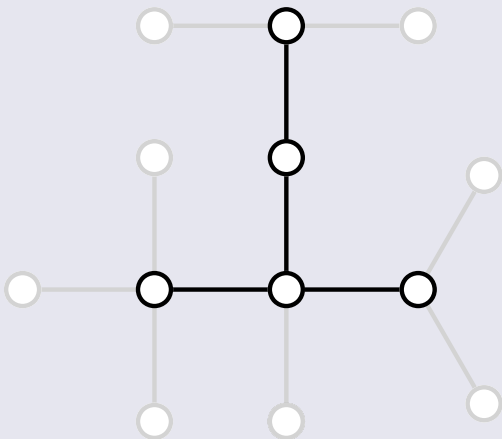


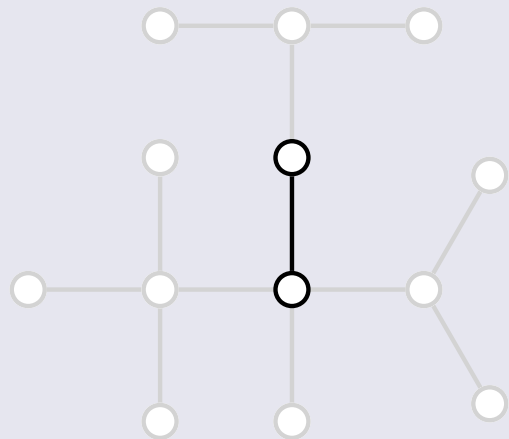


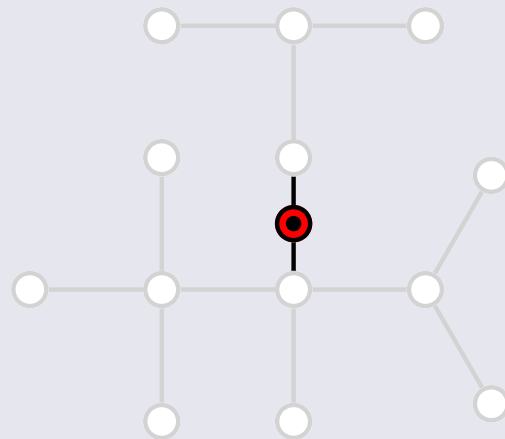
Příklad

Určete centrum stromu T_2.









Všimněte si: přidali jsme nový vrchol (a dvě hrany místo jedné).

Kořen a centrum stromů

Kořen stromu může být libovolný vrchol stromu; kořen nemusí ležet v centru.

Budeme-li chtít nějakému stromu přiřadit kořen *jednoznačně*, je nejlepší za kořen zvolit *centrum* stromu (protože centrum je určeno jednoznačně).

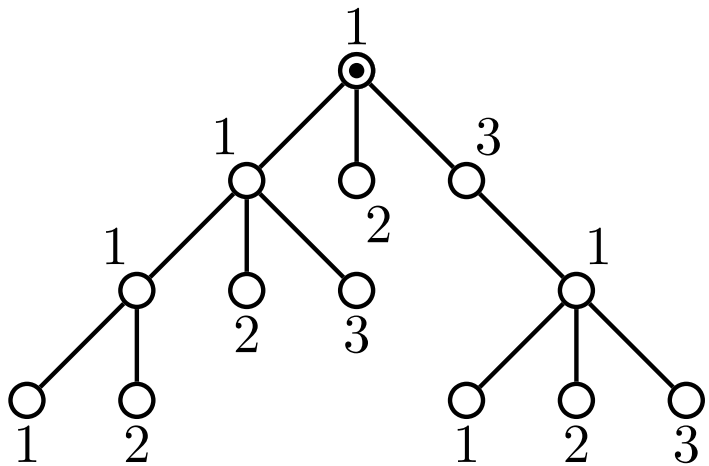
V případě, že centrem stromu je hrana uv, „přidáme“ na tuto hranu nový vrchol a zvolíme jej kořenem.

Pěstované stromy

Další informací vázanou ke kořenovým stromům bývá uspořádání potomků každého vrcholu (například seřazení potomků jedné generace v rodokmenu podle data narození).

Definice

Kořenový strom (T, r) je uspořádaný, jestliže pro každý jeho vrchol je jednoznačně dáno pořadí jeho potomků (například „zleva doprava“). Uspořádaný kořenový strom se také nazývá pěstovaný strom.



Formálně: pěstovaný strom (T, r, f), kde T je strom a r jeho kořen. Funkce $f : V(T) \to \mathbb{N}$ přitom přiřadí každému vrcholu jeho pořadí mezi sourozenci $1, 2, \ldots, k$.

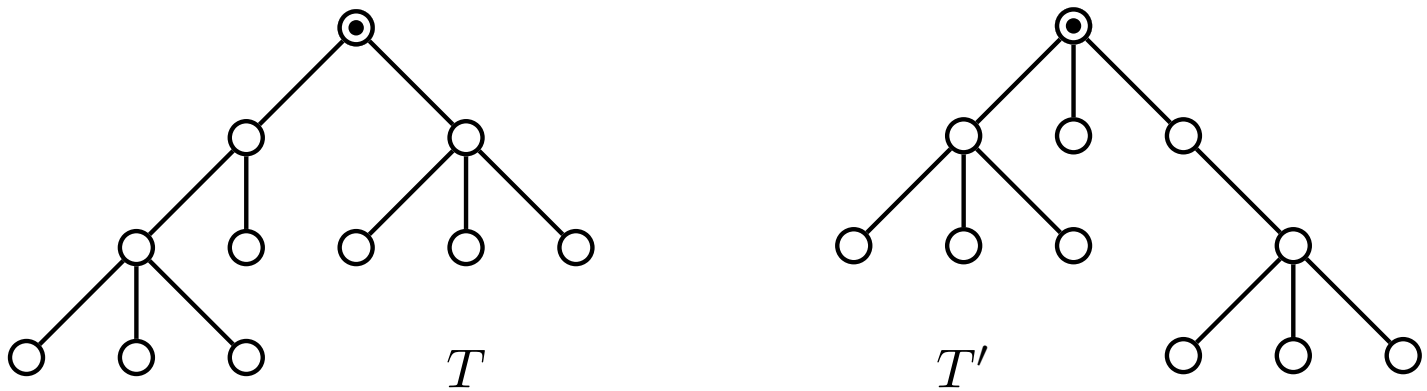
Isomorfismus stromů

Pojem isomorfismus stromů je speciálním případem isomorfismu grafů. Dva stromy jsou isomorfní, pokud jsou isomorfní jako grafy.

Připomeňme, že pro úlohu rozhodnout, zda dva obecné grafy jsou isomorfní není znám žádný rychlý algoritmus. Stromy jsou natolik speciální třída grafů, že pro ně takový algoritmus existuje! Nejprve zavedeme několik pojmů.

Definice

Dva kořenové stromy (T, r) a (T', r') jsou isomorfní pokud existuje isomorfismus mezi stromy T a T', který zobrazí kořen r na kořen r'.

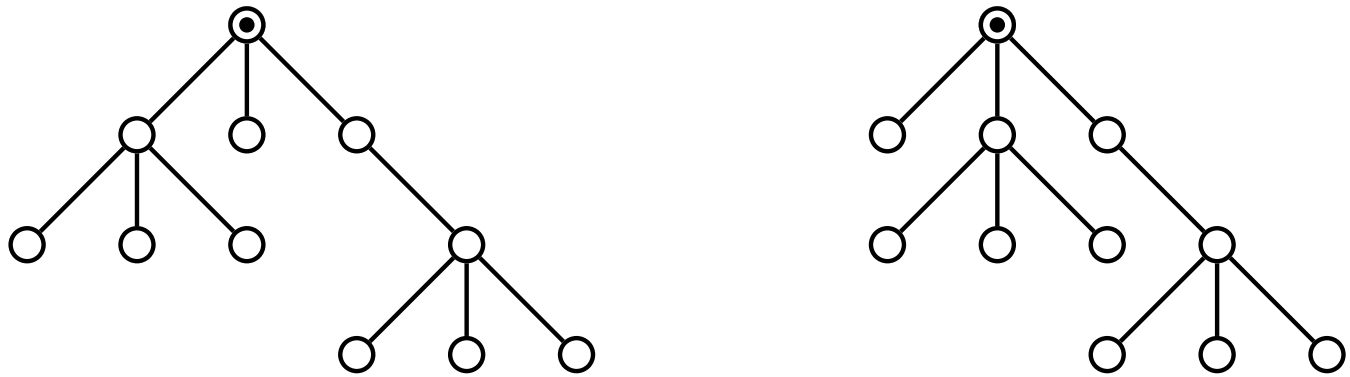


Isomorfní stromy, které nejsou isomorfní jako kořenové stromy.

Všimněte si: liší se kořen stromů T a T'.

Definice

Dva uspořádané kořenové stromy (pěstované stromy) jsou isomorfní, jestliže pro ně existuje isomorfismus kořenových stromů, který navíc zachová pořadí potomků každého vrcholu.



Isomorfní kořenové stromy, které nejsou isomorfní jako pěstované.

Všimněte si: liší se pořadí potomků kořene.

Definice

Podstromem vrcholu u daného kořenového stromu (T, r) rozumíme každou komponentu grafu $T - u$, která obsahuje některého potomka x vrcholu u.

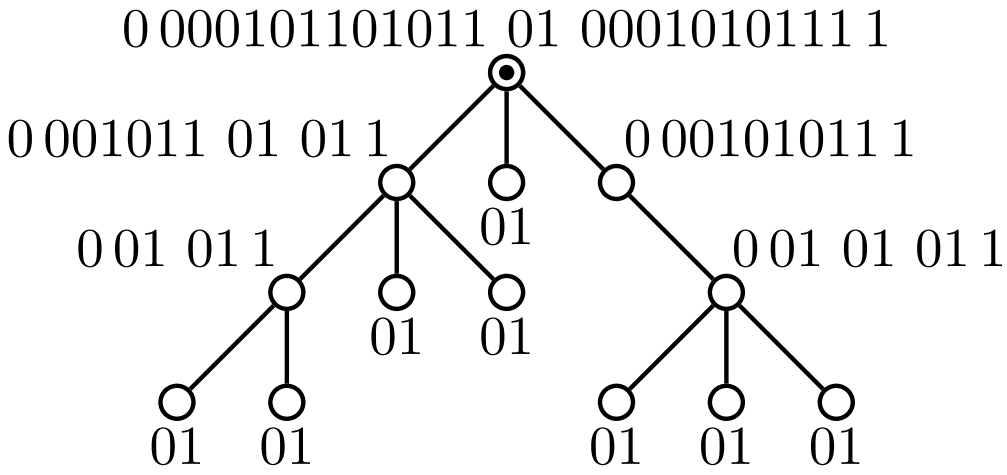
Každý podstrom vrcholu u je opět (kořenovým) stromem.

Kódování uspořádaných kořenových stromů

Každému uspořádanému kořenovému stromu lze snadno přiřadit řetězec, který jej jednoznačně popisuje. Vystačíme se symboly 0 a 1 (binární kód).

Definice

Kód uspořádaného kořenového stromu se sestaví rekurzivně z kódů všech podstromů kořene, seřazených ve stejném pořadí jako jsou seřazeny kořeny podstromů (jeho potomci), a uzavřených do páru 0 a 1 (viz Obrázek).



Kódování uspořádaného kořenového (pěstovaného) stromu.

Poznámka

Místo „0“ a „1“ lze použít jiné symboly, třeba „(“ a „)“ nebo „A“ a „B“.

Uspořádaný kořenový strom daný kódem

Ukázali jsme, jak každému uspořádanému stromu přiřadit kód.
Nyní ukážeme opačný postup: jak nakreslit uspořádaný kořenový strom s předepsaným kódem.

Lemma

Máme dán kód S uspořádaného kořenového stromu. Příslušný strom nakreslíme následujícím postupem:

- při přečtení prvního znaku „0“ na začátku položíme tužku na papír a nakreslíme kořen,
- při každém dalším přečtení znaku „0“ nakreslíme (napravo od předchozích) hranu a nového následujícího potomka x současného vrcholu a přesuneme se do x,
- při každém přečtení znaku „1“ se vrátíme do rodiče současného vrcholu, případně ukončíme kreslení a zvedneme tužku, pokud jsme v kořenu.

Pozor: ne každá posloupnost 0 a 1 je kódem nějakého stromu (více na cvičení).

Minimální kód

Kódy můžeme chápat jako řetězce a ty umíme seřadit jednoznačně, například lexikograficky.

Předpokládáme, že znak 0 je ve slovníku před znakem 1.
Například řetězec 000111 bude ve slovníku před řetězci 001011, 0011 i 01.

Je nutno rozlišovat *kód uspořádaného kořenového stromu* a *minimální kód kořenového stromu*:

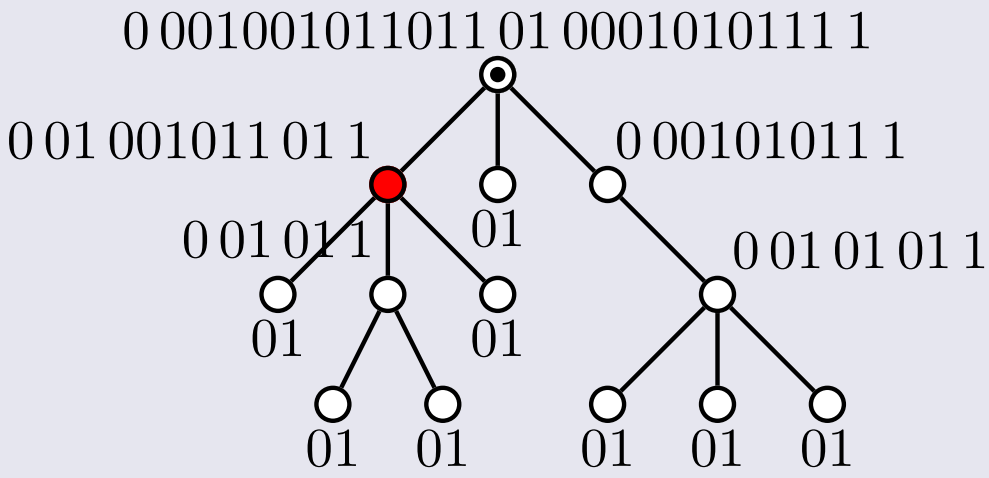
- nakreslíme-li strom, který je určen kódem uspořádaného kořenového stromu, dostaneme opět (T, r),
- nakreslíme-li strom, který je určen minimálním kódem kořenového stromu, může se pořadí potomků oproti výchozímu kořenovému stromu (T, r) lišit.

Říkáme, že jsme kořenový strom (T, r) „přepěstovali“.

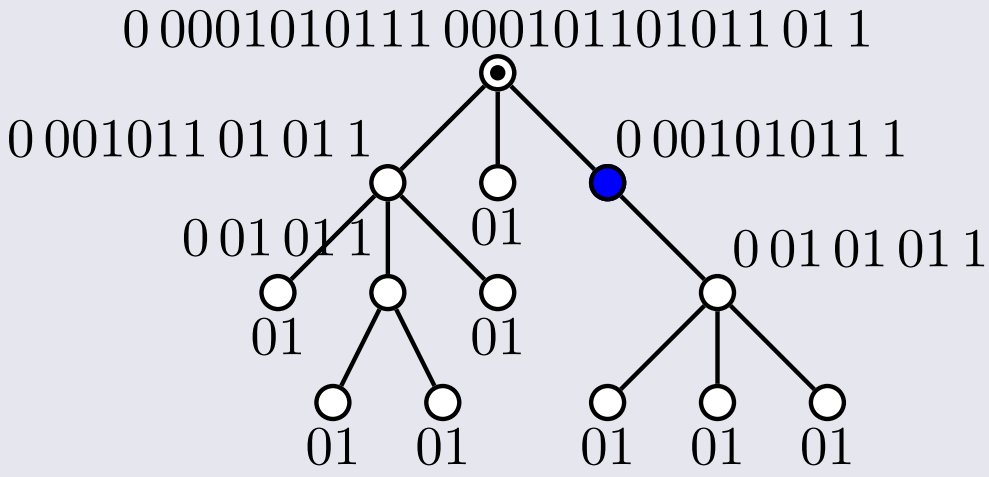
Poznámka

Horní odhad složitosti algoritmu pro nalezení minimálního kódu jednoho stromu je $O(n^3)$.

Příklad



Kód uspořádaného kořenového stromu.



Minimální kód kořenového stromu.

Při určování isomorfismu obecných stromů:

- najdeme centrum každého stromu,
- v centru volíme kořen,
- sestavíme minimální kód (pro každý vrchol seřadíme kódy potomků zleva doprava lexikograficky vzestupně podle jejich kódů),
- využijeme následující Lemma pro jednoznačnou reprezentaci minimálním kódem.

Lemma

Dva uspořádané kořenové (pěstované) stromy jsou isomorfní právě tehdy, když jejich kódy získané podle předchozího popisu jsou shodné řetězce.

Dostaneme algoritmus, který je popsán dále.

Algoritmus Určení isomorfismu stromů

Algoritmus zjišťuje, zda dané dva stromy T a U jsou isomorfní ($T \simeq^{?} U$)

```
// Máme dva stromy U, T se stejným počtem vrcholů.
Vstup < stromy T a U;
for (X=T,U) {
   // určení center daných stromů U, T
   x = centrum(X);
   if (x je jeden vrchol)
      r = x;
   else
      do X přidej vrchol r, nahrad' hranu uv hranami ru, rv;
   k[X] = minimalni_kod(X,r);
}
if ((|V(T)|==|V(U)|) && (k[T]==k[U] jako řetězce))
   vypiš("Stromy T, U jsou isomorfní.");
else
   vypiš("Stromy T, U nejsou isomorfní.");
exit;
```

Algoritmus ... pokračování (nalezení minimálního kódu)

Funkce `minimalni_kod(X,r)` najde pro strom X s kořenem r (lexikograficky) minimální kód.

```
// na vstupu je kořenový strom (případně podstrom)
vstup < kořenový strom (X,r);

funkce  minimalni_kod(strom X, vrchol r) {
   if (X má jeden vrchol)
      return "01";
   Y[1...d] = {podstromy po odebrání kořene r};
   s[1...d] = {kořeny podstromů Y[] v odpovídajícím pořadí};
                // kořeny jsou potomci kořene r
   for (i=1,...,d)
       k[i] = minimalni_kod(Y[i],s[i]);
   sort lexikograficky podle klíče k[1] <= k[2] <=...<= k[d];
   return "0"+k[1]+...+k[d]+"1";
}
```

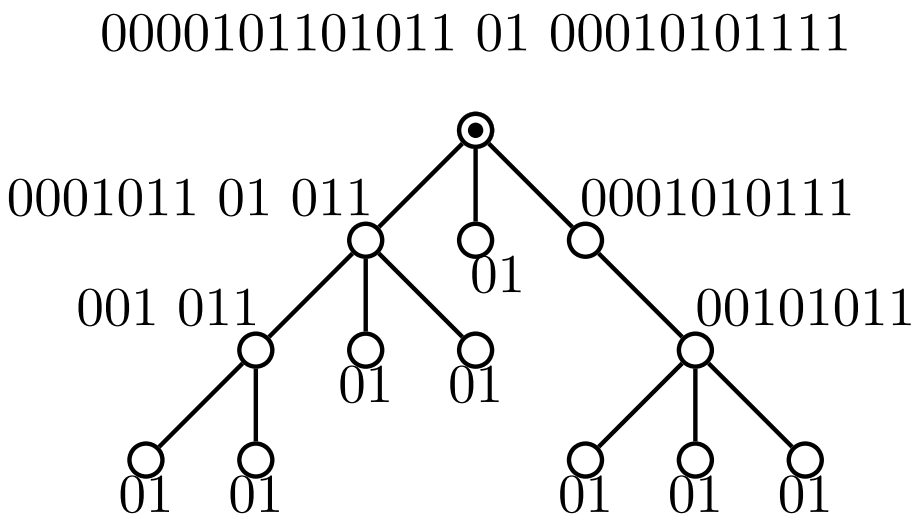
Funkce hledá minimální kód rekurzivně.

Poznámka

Všimněte si, že v Algoritmu kontrolujeme, zda oba stromy mají stejný počet vrcholů.
Například cesty P_{2n} a P_{2n+1} jistě nejsou isomorfní, ale protože centrum cesty P_{2n} je „prostřední“ hrana, tak při nalezení centra bude na tuto hranu přidán nový vrchol a vznikne tak druhá cesta P_{2n+1}. Bez porovnání počtu vrcholů by algoritmus mohl v některých případech dát chybnou odpověď.

Otázky

- Je následující kód minimální? Proč?
- Jak vypadá minimální kód?



Kostra grafu

Budeme používat následující pojmy:

- podgraf, faktor
- souvislý a nesouvislý graf
- vážený graf, ohodnocení grafu

Definice

Kostrou souvislého grafu G rozumíme takový faktor grafu G, který je stromem (faktor obsahuje všechny vrcholy grafu G).
Váhou kostry ohodnoceného grafu G rozumíme součet ohodnocení všech hran kostry.

Budeme říkat „ohodnocení hrany“ nebo „váha hrany“ a „váha kostry“.

Význam „koster“ spočívá v jejich minimalitě vzhledem k počtu hran, přičemž je zachována souvislost grafu (Věta o minimálním souvislém podgrafu).

Váhy hran (a tedy váhy koster) se mohou lišit, dostáváme:

Problém minimální kostry (MST Problem)

Je dán souvislý ohodnocený graf G s nezáporným ohodnocením hran $w : E(G) \to \mathbb{R}_0^+$. Problém minimální kostry znamená najít takovou kostru T v grafu G, která má nejmenší možnou váhu.
Formálně

$$MST = \min_{\text{kostra } T \subseteq G} \left(\sum_{e \in E(T)} w(e) \right).$$

MST (z anglického „Minimum spanning tree“) je číslo, které udává váhu kostry s nejmenší možnou vahou.

Otázky

- Kolik hran má kostra souvislého grafu na n vrcholech?
- Má smysl hledat minimální kostru v grafu se záporným ohodnocením hran?
- Je každý souvislý faktor s minimálním ohodnocením kostrou grafu?

Uvedeme několik algoritmů pro nalezení minimální kostry daného nezáporně ohodnoceného grafu.

Algoritmus Hladový pro minimální kostru

Mějme dán souvislý ohodnocený graf G s nezáporným ohodnocením hran w. Počet hran grafu G označíme m.

- Seřadíme hrany grafu G vzestupně podle jejich ohodnocení:

$$w(e_1) \leq w(e_2) \leq \cdots \leq w(e_m).$$

- Začneme s prázdnou množinou hran $T = \emptyset$ pro kostru.
- Pro $i = 1, 2, \ldots, m$ vezmeme hranu e_i a pokud přidáním této hrany nevznikne cyklus (s příslušnými vrcholy), tak přidáme hranu e_i do T. Jinak hranu e_i „zahodíme“.
- Po zpracování všech hran obsahuje T hrany minimální kostry váženého grafu G s ohodnocením w.

Ukážeme, že algoritmus funguje správně.

Věta

Hladový algoritmus najde minimální kostru souvislého grafu.

Důkaz Sporem. Nechť T je množina hran získaná v Hladovém algoritmu. Předpokládejme, že hrany $w(e_1) \leq w(e_2) \leq \cdots \leq w(e_m)$ jsou seřazeny podle váhy. Vezměme množinu hran T_0 té minimální kostry (může jich být více se stejnou hodnotou), která se s T shoduje na co nejvíce prvních hranách. Pokud $T_0 = T$, algoritmus pracoval správně.

Předpokládejme ale, že $T_0 \neq T$ a najdeme spor. Ukážeme tak, že $T_0 \neq T$ nemůže nastat.

Označme $j > 0$ takový index, že se množiny T_0 a T shodují na prvních $j-1$ hranách $e_1, \ldots, e_{j-1}$, ale neshodují se na hraně e_j. To znamená, že $e_j \in T$, ale $e_j \notin T_0$. (Podle algoritmu nemůže nastat $e_j \notin T$ a $e_j \in T_0$.) Víme, že graf $T_0 \cup \{e_j\}$ obsahuje právě jeden cyklus C. Cyklus C však nemůže být obsažen v nalezené kostře T, a proto existuje hrana e_k v C, která $e_k \notin T$ a zároveň $k > j$. Potom však je $w(e_k) \geq w(e_j)$ podle našeho seřazení hran, a proto kostra na hranách $T' = \big(T_0 \setminus \{e_k\}\big) \cup \{e_j\}$ (vzniklá nahrazením hrany e_k hranou e_j) nemá vyšší ohodnocení (není horší) než T_0, ale shoduje se s T na více hranách! To je spor s volbou kostry T_0. □

Zmíněný hladový algoritmus pro hledání minimální kostry grafu byl popsán poprvé užitím teorie grafů Kruskalem (1956). Je však známo, že Kruskal vycházel z práce českého matematika Otakara Borůvky.

Už v roce 1926 řešil český akademik Borůvka otázku optimální stavby elektrické sítě a popsal velmi podobný algoritmus užitím matic.

Algoritmus Borůvkův algoritmus pro minimální kostru

Mějme souvislý (kladně) vážený graf G s ohodnocením hran w různými čísly.
Na začátku seřadíme hrany vzestupně podle jejich ohodnocení $w(e_1) < w(e_2) < \ldots < w(e_m)$.
Kostru začneme sestavovat tak, že přidáme hranu e_i (pro $i = 1, 2, \ldots, n$), pokud přidáním nevznikne cyklus.

Jako reakce na Borůvkovu práci vypracoval Vojtěch Jarník v roce 1929 podobný algoritmus.

Jarníkův algoritmus je ve světě známý jako Primův algoritmus z roku 1957.

Algoritmus Jarníkův algoritmus pro minimální kostru

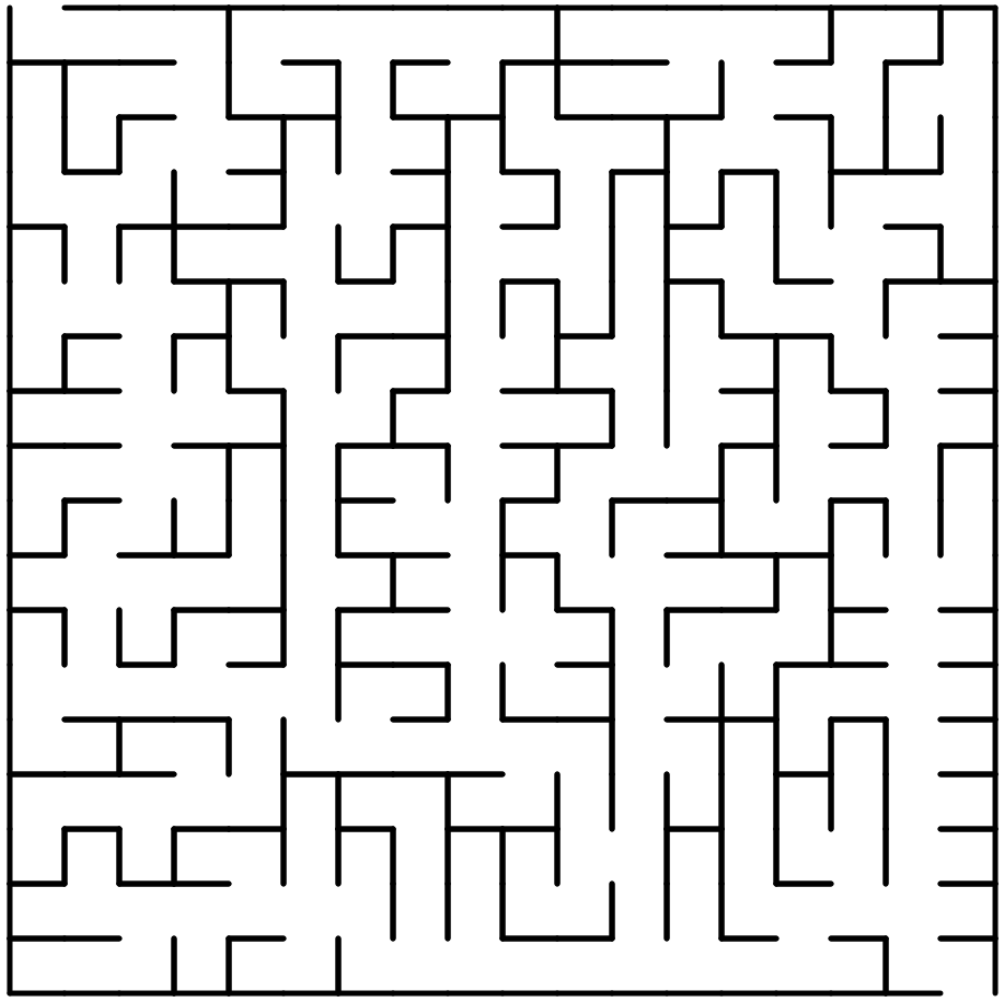
Mějme souvislý graf G s n vrcholy a s nezáporným ohodnocením hran w. Kostru začneme sestavovat z jednoho (libovolného) vrcholu. V každém kroku algoritmu přidáme nejmenší z hran, které vedou z již vytvořeného podstromu do některého ze zbývajících vrcholů grafu G.
Po $n - 1$ krocích algoritmus končí.

Všimněte si:

- v Jarníkově algoritmu hrany na začátku neseřazujeme,
- není třeba testovat vznik cyklu (přidáváme list), ušetříme čas.

Ukázky běhu hladového (Kruskalova) a Jarníkova (Primova) algoritmu najdete na adrese `http://homel.vsb.cz/~kov16/predmety_dm.php`

Pomocí algoritmů pro hledání minimální kostry snadno sestavit bludiště.



Bludiště sestavené pomocí Jarníkova algoritmu.

Podrobnosti najdete ve skriptech „Úvod do teorie grafů“.

Příště

Kapitola Barevnost a kreslení grafů

- motivace
- barevnost grafů
- rovinné nakreslení grafů
- rozpoznání rovinných grafů
- barvení map a rovinných grafů