Teorie grafů

15. přednáška z LGR

Obsah

1. **Grafové algoritmy**
   - Korektnost algoritmu
   - Složitost algoritmu
   - Representace grafu
2. **Eulerovské grafy**
   - Hledání eulerovského tahu

Grafové algoritmy

Významnou částí teorie grafů jsou grafové algoritmy, testující vlastnosti grafů nebo hledající jisté typy grafů.

Algoritmus bude orientován na problém (problem-oriented), nebudeme se zabývat psaním programu s konkrétními datovými strukturami (machine-oriented).

Algoritmus vždy představíme (případně sepíšeme v pseudokódu), dokážeme jeho korektnost a odhadneme jeho časovou složitost.

Grafové algoritmy

Korektnost algoritmu

Totální korektnost algoritmu zahrnuje dvě části:

- terminace - algoritmus zastaví pro každý přípustný vstup
- parciální korektnost - pokud algoritmus zastaví, tak poskytne očekávaný výstup

Grafové algoritmy

Korektnost algoritmu

K ověřování totální korektnosti algoritmu používáme:

- variant - veličina, která se během vykonávání algoritmu mění, a jejíž jistá hodnota zaručí, že algoritmus skončí (terminace)
- invariant - vlastnost, která se v průběhu algoritmu nemění a která při skončení zaručí správný výstup (parciální korektnost) Neměnnost invariantu se většinou dokazuje indukcí podle počtu cyklů.

Grafové algoritmy

Složitost algoritmu

Budeme odhadovat časovou a prostorovou složitost grafových algoritmů vzhledem k počtu vrcholů n a k počtu hran m daného grafu.

Bude nás zajímat pouze asymptotická složitost. Základní operace budeme odhadovat jednotkou času (což při konkrétní implementaci nemusí být odpovídající odhad).

Grafové algoritmy

Složitost algoritmu

Nechť f a g jsou reálné funkce a $g(x) \geq 0$ (stačí, aby funkce byly definované a g byla nezáporná "pro všechna dostatečně velká x").

Řekneme, že funkce f je $O(g)$, když existuje $c > 0$ a existuje $x_0 \in \mathbb{R}$ tak, že pro všechna $x \geq x_0$ je $|f(x)| \leq c\, g(x)$.

Aneb funkce f je $O(g)$, když asymptoticky neroste rychleji než konstantní násobek funkce g pro nějakou vhodnou konstantu.

Neorientované grafy

Representace grafu

Pro uložení grafu do paměti počítače můžeme zvolit různé datové struktury. Na volbě datové struktury bude pak záviset časová a prostorová náročnost grafového algoritmu.

Nechť $G = (V, E)$ je neorientovaný graf s $|V| = n$ vrcholy a $|E| = m$ hranami.
Vrcholy budeme označovat čísly $V = \{1, \ldots, n\}$.
Hrany můžeme zadat některým z následujících způsobů.

Neorientované grafy

Representace grafu

- Seznam všech hran - lze realizovat jako dvě pole délky m, např. pole U a V, přičemž hrana $e_i = \{U[i], V[i]\}$.
  Hrany mohou být částečně setříděné, např. tak, že pro každou hranu e_i uložíme její menší koncový vrchol do pole U a toto pole setřídíme vzestupně.
- Pro každý vrchol máme seznam hran incidentních s tímto vrcholem - u neorientovaného grafu je každá hrana $e = \{i, j\}$ v obou seznamech, jak pro vrchol i, tak pro vrchol j.
  Variace na toto téma - pro každý vrchol je dán seznam jeho sousedních vrcholů.

Neorientované grafy

Representace grafu

- *Matice sousednosti* je čtvercová matice $\mathbb{A} = (a_{i,j})$ typu $n \times n$ taková, že platí:

$$a_{i,j} = \begin{cases} 1, & \text{když } \{i,j\} \in E \\ 0, & \text{když } \{i,j\} \notin E \end{cases}$$

Pro obecný graf je $a_{i,j} =$ počet hran s krajními vrcholy i a j.

Matice sousednosti neorientovaného grafu je symetrická.

Variace na toto téma - matice délek hran, matice cen hran.

Neorientované grafy

Tvrzení

Nechť $\mathbb{A}$ je matice sousednosti neorientovaného grafu G.

1. Matice $\mathbb{A}^2$ má na pozici (i, i) stupeň vrcholu i.
2. Matice $\mathbb{A}^k$ má na pozici (i, j) počet sledů délky k z vrcholu i do vrcholu j.

Neorientované grafy

Representace grafu

- *Matice incidence* je obdélníková matice $\mathbb{B} = (b_{i,j})$ typu $n \times m$ taková, že platí:

$$b_{i,j} = \begin{cases} 1, & \text{jestliže } i \text{ je krajní vrchol hrany } e_j, \\ 0, & \text{v ostatních případech.} \end{cases}$$

Definice matice incidence předpokládá, že hrany jsou uspořádány. Je-li $j-$tá hrana $e_j = \{i, k\}$, pak v $j-$tém sloupci jsou jedničky právě na $i-$té a $k-$té pozici.

Neorientované grafy

Tvrzení

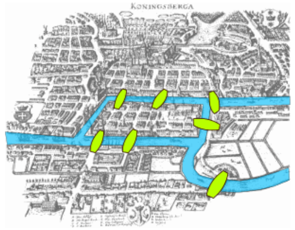
Nechť $\mathbb{B}$ je matice incidence neorientovaného grafu G s n vrcholy.

1. Graf G obsahuje kružnici, právě když jsou sloupce v matici $\mathbb{B}$ lineárně závislými vektory nad tělesem $\mathbb{Z}_2$.
2. G je souvislý graf, právě když $\operatorname{hod} \mathbb{B} = n - 1$.
3. G má p komponent souvislosti, právě když $\operatorname{hod} \mathbb{B} = n - p$.

Eulerovské grafy

Problém sedmi mostů města Královce

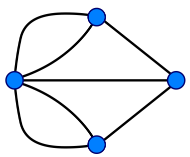
Obyvatelé pruského města Královce (Königsberg, Kaliningrad) uzavírali sázky, zda lze projít přes přes každý ze sedmi mostů právě jednou a nejlépe vrátit se tam, odkud jsme vyšli.



Eulerovské grafy

Problém sedmi mostů města Královce

Tento problém vyřešil Leonhard Euler v roce 1736 jeho práce dala vznik teorii grafů. Euler znázornil mapu města grafem, jehož vrcholy odpovídají pevninám a hrany mostům.



Problém lze nyní formulovat takto: Dá se daný graf nakreslit jedním tahem, aniž bychom nějakou hranu kreslili dvakrát? Bude to navíc uzavřený tah?

Eulerovské grafy

Definice

Eulerovský tah v neorientovaném grafu G je tah, který obsahuje všechny hrany a všechny vrcholy grafu G.

Eulerovský tah v G tedy obsahuje každou hranu grafu právě jednou (v tahu se hrany nesmí opakovat) a každý vrchol aspoň jednou.

Eulerovský tah může být uzavřený nebo otevřený.

Veškeré následující úvahy platí nejen pro obyčejné grafy, ale i pro grafy obecně - i Eulerův graf pro Královec má paralelní hrany.

Eulerovské grafy

Definice

Graf, ve kterém existuje uzavřený eulerovský tah, se nazývá *eulerovský graf*.

Tvrzení

Neorientovaný graf je eulerovský, právě když je souvislý a každý jeho vrchol má sudý stupeň.

Eulerovské grafy

Algoritmus na eulerovský tah

Vstup: souvislý graf G s vrcholy sudého stupně
Myšlenka algoritmu:

1. Vybereme libovolný vrchol v v grafu G.
2. Z vrcholu v prodlužujeme náhodně tah T pomocí dosud nepoužitých hran, dokud to jde.
3. Jestliže tah T obsahuje všechny hrany, tak skončíme. Pokud ne, pak v T existuje vrchol w, v němž začíná ještě nepoužitá hrana. Tah T ve w rozpojíme, položíme $v := w$ a opakujeme krok 2.

Výstup: tah T je uzavřený eulerovský tah v G

Eulerovské grafy

Lemma 1

Má-li graf všechny vrcholy sudého stupně, pak tah, který už nejde prodloužit (ve smyslu, že nelze přidat hrany na jeho konec), je uzavřeným tahem.

Lemma 2

Je-li graf souvislý a není-li libovolný tah v něm eulerovským tahem, pak je v tahu vrchol, který je koncovým vrcholem nějaké hrany, která neleží v tahu (= vrchol, z něhož trčí nepoužitá hrana).

Eulerovské grafy

Korektnost algoritmu

- Terminace - variant = počet nepoužitých hran, tj. těch, které nejsou v tahu T. (Ten po každém opakování kroku 2 klesá a klesne až na nulu - viz lemma 2.)
- Parciální korektnost - invariant = "Každý neprodlužitelný tah, který najdeme v kroku 2, je uzavřeným tahem."(Lze dokázat indukcí z lemmatu 1.)
  Jelikož algoritmus zastaví, až když jsou v tahu T všechny hrany, tak je T uzavřený eulerovský tah.

Eulerovské grafy

Algoritmus na eulerovský tah

Vstup: Souvislý graf $G = (V, E)$, v němž každý vrchol má kladný sudý stupeň. Označme $|E| = m$.
Výstup: Eulerovský tah $T = (v_0 e_1 v_1 e_2 \dots e_m v_m)$

Datové struktury: Tah bude representovám jako seznam, vložení jednoho tahu do druhého se provede přepsáním dvou ukazatelů. Representace grafu - pro každý vrchol v je dán seznam všech hran incidentních s vrcholem v. Nepoužitou hranu z vrcholu v lze najít v jednotkovém čase.

Eulerovské grafy

Algoritmus na eulerovský tah

(inicializace)

- vyber (náhodně) vrchol $v \in V$
- $v_0 \leftarrow v$, $T \leftarrow (v_0)$, $k \leftarrow 0$ (začneme s tahem délky $k = 0$)
- $j \leftarrow 0$ (j=délka zkontrolované části)
- všechny hrany jsou nepoužité

(přidávání hran)

- while $k < m$ (v tahu T nejsou všechny hrany) do
  - while neexistuje nepoužitá hrana z v_j do $j \leftarrow j + 1$ enddo
    (kontrola tahu - najde první vrchol v_j tahu T, z něhož trčí nepoužitá hrana)

Eulerovské grafy

- $u_0 \leftarrow v_j$, $F \leftarrow (u_0)$, $i \leftarrow 0$ (tah F délky i)
- while existují nepoužité hrany z u_i do
  - vyber nepoužitou hranu $e = \{u_i, w\}$
  - $i \leftarrow i + 1$, $f_i \leftarrow e$, $u_i \leftarrow w$ (přidej e do tahu F)
  - označ hranu e za použitou enddo

  (prodlužování tahu F po nepoužitých hranách, dokud to jde; získáme uzavřený tah $F = (u_0 f_1 u_1 \dots f_i u_i)$ z v_j do v_j)
- $T \leftarrow (v_0 e_1 \dots v_j = u_0 f_1 u_1 \dots f_i u_i = v_j \dots e_k v_k)$
- $k \leftarrow k + i$
  (vložení tahu F do tahu T v místě vrcholu v_j)
- enddo

- output $T = (v_0 e_1 \dots e_m v_m)$

Eulerovské grafy

Časová náročnost algoritmu

Časová náročnost představeného algoritmu na nalezení eulerovského tahu je $O(m)$ (lineární v závislosti na počtu hran).

Přes každou hranu "projdeme dvakrát"- jednou při přidávání do tahu T, podruhé při kontrole, zda z tahu T netrčí nepoužitá hrana. Datové struktury jsou zvoleny tak, aby nás nezdrželo ani vyhledávání nepoužité hrany z vrcholu v, ani vkládává nového tahu F do tahu T.

Eulerovské grafy

Tvrzení

V neorientovaném grafu existuje otevřený eulerovský tah, právě když je graf souvislý a má právě dva vrcholy lichého stupně (ostatní vrcholy mají sudý stupeň).

Algoritmus

Algoritmus na nalezení otevřeného eulerovského tahu je stejný jako pro uzavřený tah, pouze musíme začít ve vrcholu v lichého stupně. Inizializace: $T \leftarrow (v_0)$, kde stupeň $d(v_0)$ je lichý.

Při prvním prodlužování tahu T také skončíme ve vrcholu lichého stupně. Podgraf obsahující hrany nepoužité v tomto tahu má už stupně všech vrcholů sudé.

Eulerovské grafy

Připomeňmě, že počet vrcholů lichého stupně v grafu je vždy sudý.

Tvrzení

Nechť má souvislý neorientovaný graf G celkem $2k > 0$ vrcholů lichého stupně. Pak jej lze pokrýt k hranově disjunktními tahy.

Algoritmus

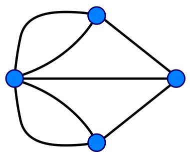
Přidáme k nových hran, kterými spojíme vrcholy lichých stupňů (libovolně do dvojic). Najdeme uzavřený eulerovský tah a přidané hrany opět vyhodíme. Uzavřený tah se rozpadne na k tahů.

Přidáním hran mohou vzniknout paralelní hrany, což nevadí.

Eulerovské grafy

Problém sedmi mostů města Královce

Jaké je tedy řešení problému sedmi mostů? Lze tento graf nakreslit jedním (uzavřeným) tahem?



Eulerovské grafy

Literatura

- J. Demel: Grafy a jejich aplikace, Academia, 2015.
- J. Matoušek, J. Nešetřil: Kapitoly z diskrétní matematiky, Nakladatelství Karolinum, 2000.
- M. Dostál: Cvičení k přednášce LGR (najdete v nich důkazy některých tvrzení z přednášky a mnoho dalších příkladů).