Paralelné algoritmy, časť č. 6

František Mráz

Kabinet software a výuky informatiky, MFF UK, Praha

Paralelné algoritmy, 2011/2012

Obsah

Reprezentácia grafu

graf $G = (V, E)$. kde $V = \{1, \ldots, n\}$ je množina vrcholov, $E \subseteq V \times V$ je množina orientovaných hrán

a) **matica susednosti:** $A[i, j] = true$ práve vtedy, keď $(i, j) \in E$

Reprezentácia grafu

graf $G = (V, E)$. kde $V = \{1, \ldots, n\}$ je množina vrcholov, $E \subseteq V \times V$ je množina orientovaných hrán

a) **matica susednosti:** $A[i, j] = true$ práve vtedy, keď $(i, j) \in E$
b) **zoznamy susednosti:** pre každý vrchol spojový zoznam jeho susedov; ukazovatele na začiatky zoznamov sú v poli s indexami $1, \ldots, n$.
Vždy predpokladáme, že môžeme prideliť procesor každému záznamu o susedovi;
$next(i, j)$ je následník záznamu o hrane (i, j) v zozname susednosti vrcholu i

Outline

1. Grafové algoritmy
   - Technika Eulerových cyklov na stromoch
   - Najkratšie cesty v grafe

Eulerov cyklus v strome

Tarjan, Vishkin 1984

Lemma

Orientovaný graf $G = (V, E)$ *má Eulerov cyklus práve vtedy, keď* G *je súvislý a pre každý vrchol* $v \in V$ *platí* $indeg(v) = outdeg(v)$.

- neorientovaný strom – v zoznamoch susednosti každá hrana je reprezentovaná dvomi šípkami opačne orientovanými $\Rightarrow$ Eulerovský graf

Eulerov cyklus v strome

Tarjan, Vishkin 1984

Lemma

Orientovaný graf $G = (V, E)$ má Eulerov cyklus práve vtedy, keď G je súvislý a pre každý vrchol $v \in V$ platí $indeg(v) = outdeg(v)$.

- neorientovaný strom – v zoznamoch susednosti každá hrana je reprezentovaná dvomi šípkami opačne orientovanými $\Rightarrow$ Eulerovský graf
- počet záznamov v zoznamoch susednosti $2(n-1)$

Eulerov cyklus v strome

Konštrukcia Eulerovho cyklu

- skonštruujeme Eulerov cyklus
  1. každý zoznam susednosti zacyklíme – nájdeme posledný prvok (zdvojovaním) a prepojíme ho na prvý prvok zoznamu – čas $O(\log n)$ s $O(n)$ procesormi

Eulerov cyklus v strome

Konštrukcia Eulerovho cyklu

- skonštruujeme Eulerov cyklus
  1. každý zoznam susednosti zacyklíme – nájdeme posledný prvok (zdvojovaním) a prepojíme ho na prvý prvok zoznamu – čas $O(\log n)$ s $O(n)$ procesormi
  2. následník v Eulerovom cykle po hrane (i, j) bude $tournext(i, j) := next(j, i)$

Eulerov cyklus v strome

Konštrukcia Eulerovho cyklu

- skonštruujeme Eulerov cyklus
  1. každý zoznam susednosti zacyklíme – nájdeme posledný prvok (zdvojovaním) a prepojíme ho na prvý prvok zoznamu – čas $O(\log n)$ s $O(n)$ procesormi
  2. následník v Eulerovom cykle po hrane (i, j) bude $tournext(i, j) := next(j, i)$
  3. for all $(i, j) \in E$ in parallel do $tournext(i, j) := next(j, i)$
     Ako implementovať?
     v čase $O(1)$ s $O(n)$ procesormi!

Zakorenenie stromu

- Eulerov cyklus “pretrhneme” v nejakom vrchole a dostaneme strom prehľadávania do hĺbky – presnejšie zoznam prehľadávania stromu

Zakorenenie stromu

- Eulerov cyklus “pretrhneme” v nejakom vrchole a dostaneme strom prehľadávania do hĺbky – presnejšie zoznam prehľadávania stromu
- očíslujeme hrany v zozname prehľadávania od 1:

$rank(r, s)$ je poradové číslo hrany (r, s) v Eulerovej ceste

Zakorenenie stromu

- Eulerov cyklus “pretrhneme” v nejakom vrchole a dostaneme strom prehľadávania do hĺbky – presnejšie zoznam prehľadávania stromu
- očíslujeme hrany v zozname prehľadávania od 1:

  $rank(r, s)$ je poradové číslo hrany (r, s) v Eulerovej ceste

- neorientovaná hrana $\{r, s\}$ sa vyskytuje ako (r, s) a (s, r), hranu s menším číslom označíme ako postupovú, druhú ako návratovú

Zakorenenie stromu

- Eulerov cyklus “pretrhneme” v nejakom vrchole a dostaneme strom prehľadávania do hĺbky – presnejšie zoznam prehľadávania stromu
- očíslujeme hrany v zozname prehľadávania od 1:

  $rank(r, s)$ je poradové číslo hrany (r, s) v Eulerovej ceste

- neorientovaná hrana $\{r, s\}$ sa vyskytuje ako (r, s) a (s, r), hranu s menším číslom označíme ako postupovú, druhú ako návratovú
- pre každú postupovú hranu (i, j) priradíme $father(j) := i$

Zakorenenie stromu

- Eulerov cyklus "pretrhneme" v nejakom vrchole a dostaneme strom prehľadávania do hĺbky – presnejšie zoznam prehľadávania stromu
- očíslujeme hrany v zozname prehľadávania od 1:

$rank(r, s)$ je poradové číslo hrany (r, s) v Eulerovej ceste

- neorientovaná hrana $\{r, s\}$ sa vyskytuje ako (r, s) a (s, r), hranu s menším číslom označíme ako postupovú, druhú ako návratovú
- pre každú postupovú hranu (i, j) priradíme $father(j) := i$
- množina postupových hrán tvorí vonkajší strom ($indeg(v) = 1$)

Zakorenenie stromu

- Eulerov cyklus "pretrhneme" v nejakom vrchole a dostaneme strom prehľadávania do hĺbky – presnejšie zoznam prehľadávania stromu
- očíslujeme hrany v zozname prehľadávania od 1:

$rank(r, s)$ je poradové číslo hrany (r, s) v Eulerovej ceste

- neorientovaná hrana $\{r, s\}$ sa vyskytuje ako (r, s) a (s, r), hranu s menším číslom označíme ako postupovú, druhú ako návratovú
- pre každú postupovú hranu (i, j) priradíme $father(j) := i$
- množina postupových hrán tvorí vonkajší strom ($indeg(v) = 1$)
- množina návratových hrán tvorí vnútorný strom ($outdeg(v) = 1$)

Počítanie v strome

- **počet potomkov** vrcholu v (vrátane v)
  for all i **in parallel do**

$$nd(i) := \frac{rank(i, father(i)) - rank(father(i), i) + 1}{2}$$

čas $O(\log n)$ s $O(n)$ procesormi

Počítanie v strome

- **počet potomkov** vrcholu v (vrátane v)
  for all i **in parallel do**

  $$nd(i) := \frac{rank(i, father(i)) - rank(father(i), i) + 1}{2}$$

  čas $O(\log n)$ s $O(n)$ procesormi
- preorderové očísľovanie vrcholov:

Počítanie v strome

- **počet potomkov** vrcholu v (vrátane v)
  for all i **in parallel do**

  $$nd(i) := \frac{rank(i, father(i)) - rank(father(i), i) + 1}{2}$$

  čas $O(\log n)$ s $O(n)$ procesormi
- preorderové očísľovanie vrcholov:
  - spočítame podzoznam postupových hrán

Počítanie v strome

- **počet potomkov** vrcholu v (vrátane v)
  for all i **in parallel do**

  $$nd(i) := \frac{rank(i, father(i)) - rank(father(i), i) + 1}{2}$$

  čas $O(\log n)$ s $O(n)$ procesormi
- preorderové očísľovanie vrcholov:
  - spočítame podzoznam postupových hrán
  - vrcholy očíslujeme – vrchol j bude mať číslo $preorder(j)$:= číslo hrany $(father(j), j) + 1$, vrchol, v ktorom Eulerova cesta začína bude mať číslo 1

Počítanie v strome

- **počet potomkov** vrcholu v (vrátane v)
  for all i **in parallel do**

  $$nd(i) := \frac{rank(i, father(i)) - rank(father(i), i) + 1}{2}$$

  čas $O(\log n)$ s $O(n)$ procesormi
- preorderové očísľovanie vrcholov:
  - spočítame podzoznam postupových hrán
  - vrcholy očíslujeme – vrchol j bude mať číslo $preorder(j)$:= číslo hrany $(father(j), j) + 1$, vrchol, v ktorom Eulerova cesta začína bude mať číslo 1
- postorderové očíslovanie analogicky

Počítanie v strome

- i je predkom $j \Leftrightarrow preorder(i) \leq preorder(j) < preorder(i) + nd(i)$
  ak máme *preorder*, tak v čase $O(1)$ s 1 procesorom

Počítanie v strome

- i je predkom $j \Leftrightarrow \mathit{preorder}(i) \leq \mathit{preorder}(j) < \mathit{preorder}(i) + \mathit{nd}(i)$
  ak máme *preorder*, tak v čase $O(1)$ s 1 procesorom
- výpočty na všetkých podstromoch

Počítanie v strome

- i je predkom $j \Leftrightarrow preorder(i) \leq preorder(j) < preorder(i) + nd(i)$
  ak máme *preorder*, tak v čase $O(1)$ s 1 procesorom
- výpočty na všetkých podstromoch
  - nech $M(i)$ je ohodnotenie vrcholu i

Počítanie v strome

- i je predkom $j \Leftrightarrow preorder(i) \leq preorder(j) < preorder(i) + nd(i)$
  ak máme *preorder*, tak v čase $O(1)$ s 1 procesorom
- výpočty na všetkých podstromoch
  - nech $M(i)$ je ohodnotenie vrcholu i
  - prvky $M(preorder(i)), \ldots, M(preorder(i) + nd(i) - 1)$ sú ohodnotenia vrcholov, ktoré sú potomkami vrcholu i

Počítanie v strome

- i je predkom $j \Leftrightarrow preorder(i) \leq preorder(j) < preorder(i) + nd(i)$
  ak máme *preorder*, tak v čase $O(1)$ s 1 procesorom
- výpočty na všetkých podstromoch
  - nech $M(i)$ je ohodnotenie vrcholu i
  - prvky $M(preorder(i)), \ldots, M(preorder(i) + nd(i) - 1)$ sú ohodnotenia vrcholov, ktoré sú potomkami vrcholu i
  - minimá v každom podstrome sa spočítajú metódou miním na intervaloch

Počítanie v strome

- i je predkom $j \Leftrightarrow preorder(i) \leq preorder(j) < preorder(i) + nd(i)$ ak máme *preorder*, tak v čase $O(1)$ s 1 procesorom
- výpočty na všetkých podstromoch
  - nech $M(i)$ je ohodnotenie vrcholu i
  - prvky $M(preorder(i)), \ldots, M(preorder(i) + nd(i) - 1)$ sú ohodnotenia vrcholov, ktoré sú potomkami vrcholu i
  - minimá v každom podstrome sa spočítajú metódou miním na intervaloch
  - čas $O(\log n)$ s $O(n)$ procesormi

Outline

Dĺžka najkratších ciest v grafe

Kučera Luděk

Vstup: M matica ohodnotenia hrán grafu $G = (V, E)$, $M[i, j] \geq 0$ pre všetky $i, j \in V$

Výstup: matica M'

$M'[i, j] = 0$ pre $i = j$

$M'[i, j] = \min\{M[i_0, i_1] + M[i_1, i_2] + \cdots + M[i_{k-1}, i_k]\}$

minimum sa počíta cez všetky postupnosti $i_0, i_1, \ldots, i_k$, pre ktoré $i_0 = i$ a $i_k = j$

Dĺžka najkratších ciest v grafe

Kučera Luděk

for all i, j **in parallel do** $m[i,j] := M[i,j]$
repeat $\log n$ **times** **begin**
  for all i, j, k **in parallel do**
    $q[i,j,k] := m[i,j] + m[j,k]$;
  for all i, j **in parallel do**
    $m[i,j] := \min\{m[i,j], q[i,1,j], q[i,2,j], \ldots, q[i,n,j]\}$;
end;
for all i, j **in parallel do**
  if $i \neq j$ **then** $M'[i,j] := m[i,j]$
    else $M'[i,j] := 0$

Dĺžka najkratších ciest v grafe

Kučera Luděk

for all i, j **in parallel do** $m[i,j] := M[i,j]$
repeat $\log n$ **times** **begin**
  for all i, j, k **in parallel do**
    $q[i,j,k] := m[i,j] + m[j,k]$;
  for all i, j **in parallel do**
    $m[i,j] := \min\{m[i,j], q[i,1,j], q[i,2,j], \ldots, q[i,n,j]\}$;
end;
for all i, j **in parallel do**
  if $i \neq j$ **then** $M'[i,j] := m[i,j]$
    else $M'[i,j] := 0$

- po k-tej iterácii repeat-cyklu je $m[i,j]$ dĺžka najkratšej cesty vedúcej cez maximálne 2^k hrán

Dĺžka najkratších ciest v grafe

Kučera Luděk

for all i, j **in parallel do** $m[i,j] := M[i,j]$
repeat $\log n$ **times** **begin**
  for all i, j, k **in parallel do**
    $q[i,j,k] := m[i,j] + m[j,k]$;
  for all i, j **in parallel do**
    $m[i,j] := \min\{m[i,j], q[i,1,j], q[i,2,j], \ldots, q[i,n,j]\}$;
end;
for all i, j **in parallel do**
  if $i \neq j$ **then** $M'[i,j] := m[i,j]$
    else $M'[i,j] := 0$

- po k-tej iterácii repeat-cyklu je $m[i,j]$ dĺžka najkratšej cesty vedúcej cez maximálne 2^k hrán
- zložitosť

Dĺžka najkratších ciest v grafe

Kučera Luděk

for all i, j **in parallel do** $m[i,j] := M[i,j]$
repeat $\log n$ **times** **begin**
  for all i, j, k **in parallel do**
    $q[i,j,k] := m[i,j] + m[j,k]$;
  for all i, j **in parallel do**
    $m[i,j] := \min\{m[i,j], q[i,1,j], q[i,2,j], \ldots, q[i,n,j]\}$;
end;
for all i, j **in parallel do**
  if $i \neq j$ **then** $M'[i,j] := m[i,j]$
  else $M'[i,j] := 0$

- po k-tej iterácii repeat-cyklu je $m[i,j]$ dĺžka najkratšej cesty vedúcej cez maximálne 2^k hrán
- zložitosť

CREW čas $O(\log^2 n)$ s $O(\frac{n^3}{\log n})$ procesormi – výpočet miním v čase $O(\log n)$ s $O(\frac{n}{\log n})$ procesormi

Dĺžka najkratších ciest v grafe

Kučera Luděk

for all i, j **in parallel do** $m[i, j] := M[i, j]$
repeat $\log n$ **times** **begin**
  for all i, j, k **in parallel do**
    $q[i, j, k] := m[i, j] + m[j, k]$;
  for all i, j **in parallel do**
    $m[i, j] := \min\{m[i, j], q[i, 1, j], q[i, 2, j], \ldots, q[i, n, j]\}$;
end;
for all i, j **in parallel do**
  if $i \neq j$ **then** $M'[i, j] := m[i, j]$
    else $M'[i, j] := 0$

- po k-tej iterácii repeat-cyklu je $m[i, j]$ dĺžka najkratšej cesty vedúcej cez maximálne 2^k hrán
- zložitosť

CREW čas $O(\log^2 n)$ s $O(\frac{n^3}{\log n})$ procesormi – výpočet miním v čase $O(\log n)$ s $O(\frac{n}{\log n})$ procesormi

COMMON čas $O(\log n)$ s $O(n^4)$ procesormi – výpočet miním v čase $O(1)$ s $O(n^2)$ procesormi