Algoritmy a datové struktury I

10. cvičení

čtvrtek 28. 4. 2016 9:00

Nejkratší cesty

- BFS pro neohodnocené grafy.
- NP-úplné pro grafy se zápornými cykly – ohodnotíme všechny hrany -1 a pak nejkratší cesta mezi dvěma vrcholy je Hamiltonovská cesta (tj. cesta délky $1-n$), pokud existuje.
- Pro grafy bez záporných cyklů existuje nejkratší sled, který je cestou.

Relaxační algoritmy

- Pro každý vrchol udržujeme $d(v)$ délku nejkratšího sledu, pomocí kterého se zatím umíme do v dostat.
- Odhad na vzdálenost upravujeme voláním funkce `relax`.
- Otevřené jsou ty vrcholy, které ještě musíme zrelaxovat. Na počátku je otevřený jen v_0, $d(v_0) = 0$ a $d(v) = \infty$ pro $v \neq v_0$.

Function `relax`(u)
  for $\forall v : uv \in E$ **do**
    if $d(u) + w(uv) < d(v)$ **then**
      $d(v) \leftarrow d(u) + w(uv)$
      $stav(v) \leftarrow$ otevřený
  $stav(u) \leftarrow$ uzavřený

- Pokud graf neobsahuje záporné cykly, tak proces relaxací skončí, ať vrcholy vybíráme jakkoliv. Při nevhodném pořadí může trvat exponenciálně dlouho.
- Když už nelze žádný vrchol relaxovat, tak $d(v)$ jsou délky nejkratších cest.
- Dijkstrův algoritmus ($\mathcal{O}(m \log_{m/n} n)$ s $\frac{m}{n}$-regulární a $\mathcal{O}(n \log n + m)$ s Fibonacciho haldou) – relaxuji vrchol s nejmenší vzdáleností.
- Bellmanův-Fordův algoritmus ($\mathcal{O}(mn)$) – vrcholy k relaxaci dávám do fronty. Funguje se uzápornými hranami.

d-regulární haldy

- Podobné binární haldě, ale každý uzel má d potomků.
- Stejně jako binární haldu ji lze implementovat v poli (potomci na indexech $di + 1$ až $di + d$).
- Insert a Decrease trvají $\mathcal{O}(\log_d n)$.
- ExtractMin a Increase trvají $\mathcal{O}(d \log_d n)$.

Floydův-Warshallův algoritmus

- Spočte matici vzdáleností D^n v čase $\Theta(n^3)$ s pamětí $\Theta(n^2)$. Funguje se zápornými hranami.
- Postupně počítá matice D^k, kde $D^k_{u,v}$ je délka nejkratšího uv-sledu, jehož vnitřní vrcholy jsou $\leq k$ (indexováno od 1).
- Platí $D^{k+1}_{u,v} = \min(D^k_{u,v}, D^k_{u,k+1} + D^k_{k+1,v})$.

Příklady

1. Mějme mapu města ve tvaru orientovaného grafu. Každou hranu ohodnotíme podle toho, jaký nejvyšší kamion po dané ulici může projet. Po cestě tedy projede maximálně tak vysoký náklad, kolik je minimum z ohodnocení jejích hran. Jak pro zadané dva vrcholy najít cestu, pro níž projede co nejvyšší náklad?
2. Silnice v mapě máme ohodnocené dvěma čísly: délkou a mýtem (poplatkem za projetí). Jak najít nejlevnější z nejkratších cest?
3. Ukažte příklad grafu s celočíselně ohodnocenými hranami, na kterém Dijkstrův algoritmus běží exponenciálně dlouho.
4. Upravte Bellmanův-Fordův algoritmus, aby uměl detekovat záporný cyklus dosažitelný z vrcholu v_0. Uměli byste tento cyklus vypsat?
5. Papeho algoritmus funguje podobně jako Bellmanův-Fordův, pouze místo fronty používá zásobník. Ukažte, že tento algoritmus v nejhorším případě běží exponenciálně dlouho.
6. Uvažujme následující algoritmus: provedeme n fází, v každé z nich postupně relaxujeme všechny vrcholy. Spočte tento algoritmus správné vzdálenosti? Jak si stojí v porovnání s Bellmanovým-Fordovým algoritmem?
7. Jak z výsledku Floydova-Warshallova algoritmu zjistíme, kudy nejkratší cesta mezi nějakými dvěma vrcholy vede?
8. Upravte Floydův-Warhsallův algoritmus, aby zjistil, zda v grafu existuje záporný cyklus.

Domácí úkoly

Úkoly jsou za plný počet bodů **3 týdny** od zadání (deadline je počátek cvičení), poté za polovinu bodů. Úkoly mi posílejte na `husek+ads@iuuk.mff.cuni.cz`.

1. Mějme mapu města, která má časem potřebným na průjezd ohodnocené nejen hrany (silnice), ale také vrcholy (křižovatky). Upravte Dijkstrův algoritmus, aby našel nejrychlejší cestu i v tomto případě. [**cesta, 9**]
2. Mějme ohodnocený orientovaný graf G. Vymyslete algoritmus, který nalezne všechny hrany, jež leží na alespoň jedné nejkratší cestě z v_0. [**nejh, 9**]
3. Opět ohodnocený orientovaný graf. Kritická hrana budiž taková, která leží na všech nejkratších cestách z v_0. Tedy ta, jejíž prodloužení by ovlivnilo vzdálenost. Navrhněte algoritmus, který najde všechny kritické hrany. [**krit, 9**]
4. Mějme ohodnocený orientovaný graf. Chceme v něm najít nejkratší cyklus ze čtyř hran v čase lepším než $\Theta(n^4)$. [**4cyk, 9**]
5. Opět ohodnocený orientovaný graf. Jsou dány vrcholy u, v a číslo k. Spočítejte, jak dlouhý je nejkratší sled z u do v tvořený k hranami. [**ksled, 9**]