6. CVIČENÍ Z ADS 1, ČTVRTEK 15:40, LS '24

Hledání minimálních koster s Jarníkem, Borůvkou a Kruskalem

1. *Rozcvička.*
a) Jak hledání minimální kostry ovlivní záporné hrany nebo záporné cykly?
b) Co kdybychom chtěli najít kostru *maximální* namísto minimální?
c) Jak se zbavit předpokladu o unikátních vahách hran?

2. *Dynamická kostra.* Máme nalezenu minimální kostru a nyní chceme najít novou, pokud:
a) z grafu odstraníme hranu,
b) do grafu přidáme hranu,
c) snížíme váhu hrany, nebo
d) zvýšíme váhu hrany.
Na změnu stačí lineární čas. (Umí se i amortizovaný čas $O(\text{poly}(\log n))$.)

3. *Kostry s malými čísly.* Vymyslete algoritmus na nalezení minimální kostry grafu, v němž jsou váhy hran přirozená čísla z $\{1, \ldots, k\}$, kde k je parametr (předpokládejte, že $k \in o(\log n)$).

Řezové lemma. Nechť G je souvislý graf s unikátním ohodnocením hran a R elementární řez v G. Pak nejlehčí hrana řezu R leží v minimální kostře.

Cyklové lemma. Nechť G je souvislý graf s unikátním ohodnocením hran a C cyklus v G. Pak nejtěžší hrana C neleží v minimální kostře.

Žluto-modrý algoritmus. Na začátku obarvíme všechny hrany černě. Dokud existuje alespoň jedna černá hrana, aplikujeme jedno z následujících pravidel:
- Zvolíme libovolně elementární řez R, jehož nejlehčí hrana e je černá, a obarvíme tuto hranu e modře.
- Zvolíme libovolně cyklus C, jehož nejtěžší hrana e je černá, a obarvíme tuto hranu e žlutě.

4. *Všichni tři jsou žluto-modří.* Ukažte, že Jarníkův, Borůvkův i Kruskalův algoritmus se dají popsat jako speciální případy žluto-modrého algoritmu.

5. *Správnost žluto-modrých algoritmů.* Dokažte cyklové lemma. Poté odvoďte, že libovolná varianta žluto-modrého algoritmu nalezne minimální kostru, která se bude skládat právě z modrých hran. (pro graf s unikátními váhami).

6. *Neunikátní lemmata.* Jak se změní řezové a cyklové lemma pro neunikátní váhy?

Bonusové úlohy:

7. *Druhá nejlehčí kostra.* Vymyslete, jak najít druhou nejlepší kostru grafu. (Nemusíte dosáhnout lineární časové složitosti.)

8. *Kontrahující Borůvka.* Borůvkův algoritmus můžeme upravit, aby každý strom lesa udržoval zkontrahovaný do jednoho vrcholu. Iterace pak vypadá tak, že si každý vrchol vybere nejlehčí incidentní hranu, tyto hrany zkontrahujeme a zapamatujeme si, že patří do minimální kostry. Ukažte, jak tento algoritmus implementovat tak, aby běžel v čase $O(m \log n)$. Jak si poradit s násobnými hranami a smyčkami, které vznikají při kontrakci?

9. *Rovinný Borůvka.* Jak rychle najde vhodně implementovaný algoritmus z předchozího příkladu minimální kostru v rovinném grafu? Naopak najděte (nerovinný) graf, na kterém algoritmus z předchozího příkladu poběží v čase $\Theta(m \log n)$.