Za domácí úkol máte příklady 1, 4 a 7.4. Důkladně si přečtěte zadání!

U všech níže uvedených příkladů se snažte najít algoritmus, který zadanou úlohu vyřeší co možná nejrychleji s co nejmenší spotřebou paměti. Vždy uvádějte časovou a paměťovou složitost vašich algoritmů i s důkazy správnosti vašich tvrzení.

Příklad 1. Tramtárii jezdí po železnici samé rychlíky, které nikde po cestě nestaví. V jízdním řádu je pro každý rychlík uvedeno počáteční a cílové nádraží, čas odjezdu a čas příjezdu. Nyní stojíme v čase t na nádraží a a chceme se co nejrychleji dostat na nádraží b. Navrhněte algoritmus, který najde takové spojení.

Příklad 2. V souvislém hranově ohodnoceném grafu najděte kostru, která má minimální součin vah hran. Jak byste hledali kostru T s minimální hodnotou $\max_{e \in T} w(e)$, kde $w(e)$ je váha hrany e.

Příklad 3. Uvažujme souvislý graf G a množinu všech jeho koster. Charakterizujte hrany, které leží ve všech kostrách.

Příklad 4. Uvažujme souvislý, hranově ohodnocený graf G a množinu všech jeho minimálních koster. Dokažte, že hrana e není v žádné minimální kostře právě tehdy, když e je nejtěžší hranou nějaké kružnice v G.

Příklad 5. Uvažujme následující algoritmus: Začněme ze souvislého grafu G a dokud existuje hrana po jejíž odstranění zůstane graf souvislý, tak ji odstraňme. Dokažte, že výsledný graf tvoří kostru grafu G.

Upravme algoritmus pro hranově ohodnocené grafy: Postupně procházejme všechny e od nejtěžší a odstraňme e z grafu, pokud po jejím odstraněním zůstane graf souvislý. Dokažte, že výsledný graf tvoří minimální kostru grafu G. Jaká je složitost tohoto algoritmu?

Příklad 6. Uvažujme následující rekurzivní algoritmus pracující na souvislém grafu G a vracející nějakou množinu hran: Rozdělme množinu vrcholů grafu na neprázdné množiny A a B a zvolme libovolnou hranu e grafu G vedoucí mezi A a B. Dvakrát rekurzivně zavolejme algoritmus na podgrafy G indukované množinami A a B a označme T_A a T_B množiny hran, která rekurze vrátí. Nakonec rekurze vrátí sjednocení $T_A \cup T_B \cup \{e\}$. Rekurze končí, když množina vrcholů je jednoprvková, a v tomto případě vrátí prázdnou množinu. Vrátí tento algoritmus kostru?

Jestliže mezi všemi hranami vedoucími mezi A a B vybereme nejlehčí hranu e, vrátí algoritmus minimální kostru?

Příklad 7. Rozhodněte, zda následující tvrzení platí:

1. Jestliže graf G má alespoň n hran, pak nejtežší hrana neleží v minimální kostře.
2. Každá hrana minimální kostry je nejlehčí hranou nejakého řezu.
3. Minimální kostra je souvislý podgraf s nejmenším součtem vah hran.
4. Nejkratší cesta mezi libovolnými dvěma vrcholy leží v nějaké minimální kostře.