Za domácí úkol máte příklady 5 a 8. Důkladně si přečtěte zadání!

Příklad 1. 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.

Definition 1 (Binární strom)**.** Strom nazveme binární, pokud je zakořeněný a každý vrchol má nejvýše dva syny, u nichž rozlišujeme, který je levý a který pravý. Vrcholy rozdělíme podle vzdálenosti od kořene do hladin: v nulté hladině leží kořen, v první jeho synové atd.

Pro vrchol v binárního stromu T značíme:
- $k(v)$ je klíč uložený ve vrcholu v.
- $T(v)$ je podstrom obsahující vrchol v a všechny jeho potomky.
- $l(v)$ a $r(v)$ jsou levý a pravý synové vrcholu v.
- $L(v)$ a $R(v)$ jsou podstromy pravého a levého vrcholu v.
- $h(v)$ je hloubka stromu $T(v)$, čili maximum z délek cest z v do listů.

Definition 2 (Binární vyhledávací strom)**.** Binární vyhledávací strom je binární strom, jehož každému vrcholu v přiřadíme unikátní klíč $k(v)$ z univerza. Přitom musí pro každý vrchol v platit:
- Kdykoliv $a \in L(v)$, pak $k(a) < k(v)$.
- Kdykoliv $a \in R(v)$, pak $k(a) > k(v)$.

Příklad 2.
- Rozhodněte, zda následující podmínka je ekvivalentní s definicí binárního vyhledávacího stromu. Pro každý vrchol v musí platit, že $k(l(v)) < k(v) < k(r(v))$.
- Rozhodněte, zda následující podmínka je ekvivalentní s definicí binárního vyhledávacího stromu. Pro každý vrchol v musí platit, že $\max\{k(a);\ a \in L(v)\} < k(v) < \min\{k(a);\ a \in R(v)\}$.

Příklad 3. Vypište všechny prvky binárního vyhledávacího stromu v setříděném pořadí.

Příklad 4. Pro danou setříděnou posloupnost prvků vytvořte binární vyhledávací strom s co nejmenší hloubkou.

Příklad 5. Jak v binárním vyhledávacím stromu najdeme daný prvek? Jestliže daný prvek se ve stromu nevyskytuje, jak najdeme nejbližší prvky (nejmenší větší a největší menší)?

Příklad 6. Pro daný binární vyhledávací strom a dva klíče x a y vyjmenujte všechny prvky mezi x a y. Dokázali byste rychle zjistit počet prvků mezi x a y?

Příklad 7. Jak si v paměti uložit posloupnost čísel $a_1, \dots, a_n$ tak, abychom dokázali rychle odpovídat na následující dotaz: Pro dané indexy i a j urči maximum z hodnot $a_i, \dots, a_j$.

Příklad 8. Jak upravit binární vyhledávací strom tak, aby uměl rychle najít k-tý prvek pro dané k?

Příklad 9. Mějme posloupnost matic $A_1, \dots, A_n$, kterou si chceme uložit tak, abychom rychle uměli odpovídat následující dotazy: Pro dané indexy i a j urči součin matic $A_i \cdots A_j$. Součin matic pomalá operace, takže chceme ukládat pomocné informace takové, že k vyhodnocení dotazu potřebujeme co nejmenší počet součinů matic. Mohli bychom si pamatovat součin matic $A_i \cdots A_j$ pro všechny indexy $i \leq j$, ale takových dvojic je $\binom{n+1}{2}$ a tolik paměti nemáme. Kolik paměti potřebujete, abyste dokázali dotaz vyhodnotit jen na jeden součin dvou matic? Kolik součinů potřebujete, pokud smíte použít jen $\mathcal{O}(n)$ paměti? Můžete předpokládat, že jednu matici lze uložit v $\mathcal{O}(1)$ paměti. Nezapomeňte, že násobení matic je sice asociativní, ale není komutativní ani nelze matice dělit.