1 Předchozí příklady

1. Dokažte, že pro kladné funkce f, g, h platí: $(f \in O(g) \wedge g \in O(h)) \implies f \in O(h)$
2. Šéf agentů: Podařilo se vám sehnat schéma sítě tajných agentů. Má podobu orientovaného grafu, jehož vrcholy jsou agenti a hrana popisuje, že jeden agent velí druhému. Kdykoliv agent obdrží rozkaz, předá ho všem agentům, kterým velí. Šéfem sítě je libovolný agent, který vydá-li rozkaz, dostanou ho časem všichni ostatní agenti. Vymyslete lineární algoritmus, jež najde šéfa sítě. Umíte najít všechny šéfy?
3. Strážníci: Mapa městečka má tvar stromu. Na křižovatky chceme rozmístit co nejméně strážníků tak, aby každá ulice byla z alespoň jedné strany hlídaná. Nalezněte řešení v lineárním čase.
4. Vysvobození robotů: Koupili jste na inzerát dvojici skvělých robotů. Lacino, neboť jsou právě uvěznění v bludišti (čtvercová síť s některými políčky blokovanými). Znáte jejich polohy a můžete jim rádiem vysílat povely pro posun o políčko na sever, jih, východ či západ, abyste je dostali na okraj bludiště. Háček je ale v tom, že na každý povel reagují oba roboti. Vymyslete algoritmus, který najde nejkratší posloupnost povelů, jež vysvobodí oba roboty. Dodejme ještě, že robot ignoruje povel, který by způsobil okamžitý náraz do zdi nebo do druhého robota, a že jakmile se robot dostane na okraj, odchytíme ho a další povely neposlouchá.
5. Vítěz: n sportovců sehrálo zápasy každý s každým. Chceme zjistit, zda existuje někdo, kdo vyhrál nad všemi ostatními. Máme-li matici výsledků zápasů již načtenou v paměti, lze to spočítat v čase $O(n)$.
6. Silně souvislá orientace: V každém neorientovaném grafu bez mostů je možné hrany zorientovat tak, aby vznikl silně souvislý orientovaný graf. Vymyslete algoritmus, který takovou orientaci najde v čase O(n+m).
7. Vymyslete algoritmus, který nalezne všechny hrany, jež leží na alespoň jedné nejkratší cestě mezi vrcholy u a v.
8. Kritická hrana budiž taková, která leží na všech nejkratších cestách. Tedy ta, jejíž prodloužení by ovlivnilo vzdálenost. Navrhněte algoritmus, který najde všechny kritické hrany. Opět řešíme pouze cestu mezi vrcholy u a v.
9. Dokažte, že projdeme-li celý strom opakovaným hledáním následníka, strávíme tím čas $\Theta(n)$.
10. Pořadí permutací: Uspořádejme všechny permutace na množině $\{1, \ldots, n\}$ lexikograficky. Vymyslete algoritmus, který pro dané k sestrojí v pořadí k-tou permutaci v čase $O(n \log n)$. Navrhněte též převod permutace na její pořadové číslo.
11. Intervalový strom: Nechť v každém vrcholu stromu kromě klíče sídlí ještě nějaká celočíselná hodnota. Upravte strom, aby uměl rychle zjistit největší hodnotu přiřazenou nějakému klíči z intervalu $[a, b]$.
12. Dokažte, že budeme-li reprezentovat množiny pomocí BVS, nelze sjednocení provést rychleji než lineárně v nejhorším případě. Platí to dokonce i tehdy, máme-li na vstupu dokonale vyvážený strom a výstup může být jakkoliv nevyvážený.