

## Sedmé cvičení ADS 1

**Příklad 1 (zoo):** V zoo jsou pouze jednosměrné chodníčky a proto lze její mapu reprezentovat jako orientovaný graf se speciálními vrcholy pro vchod a východ. Kolem chodníků jsou jednotlivá zvířata. Popularita zvířat se počítá podle počtu návštěvníků, kteří projdou po odpovídajícím chodníčku. Počet návštěvníků lze určit pomocí fotobuňek umístěných na daném chodníku. Snímače jsou ale drahé. Navrhněte algoritmus, který rozmístí po zoo co nejmenší počet fotobuňek tak, aby bylo možné ze sebraných dat večer určit pro každý chodník kolik návštěvníků po něm prošlo.

**Příklad 2 (UF):** Datová struktura pro Union-Find s polem by se dala zrychlit tím, že bychom pokaždé přečíslovali tu menší z komponent. Dokažte, že pak je během života struktury každý vrchol přečíslován nejvýše  $(\log n)$ -krát. Co z toho plyne pro složitost operací? Nezapomeňte, že je potřeba efektivně zjistit, která z komponent je menší, a vyjmenovat její vrcholy.

**Příklad 3 (cykly):** V orientovaném grafu jsou některé vrcholy obarvené zeleně. Jak zjistit, jestli existuje cyklus obsahující alespoň jeden zelený vrchol?

**Příklad 4 (následníci):** Dokažte, že projdeme-li celý binární vyhledávací strom opakovaným hledáním následníka, strávíme tím čas  $\Theta(n)$ .

**Příklad 5 (stavba BVS):** Navrhněte algoritmus, který ze setříděného pole vyrobí v lineárním čase dokonale vyvážený BVS.

**Příklad 6 (sloučení stromů):** Navrhněte algoritmus, který dostane dva BVS  $T_1$  a  $T_2$  a sloučí jejich obsah do jediného BVS v čase  $O(|T_1| + |T_2|)$ . Dokažte, že pro množiny reprezentované binárními vyhledávacími stromy není možné sloučit v lepším než lineárním čase, a to ani v případě, kdy na vstupu dostanete dokonale vyvážené stromy a výstup nemusí být vyvážený.

**Příklad 6 (rozdelení stromů):** Pro zadaný BVS  $T$  a hodnotu  $k$  navrhněte algoritmus, který  $T$  rozdělí na dva BVS  $T_1$  a  $T_2$ , přičemž v  $T_1$  jsou hodnoty menší nebo rovné  $k$  a v  $T_2$  jsou hodnoty větší než  $k$ .

**Domácí úkol 8 (AVL, 15 bodů):** Vymyslete, jak ukládat znaménka do vrcholů AVL stromu tak, abyste si vystačili s jedním bitem na vrchol. Znaménko vrcholu musí být z uložených bitů spočítat v konstantním čase.

**Domácí úkol 9 (rank, 10 bodů):** Ukažte, jak upravit vyhledávací strom, aby navíc uměl efektivně provádět operaci  $\text{Rank}(x, y)$ , která odpoví, kolik vrcholů stromu leží v intervalu  $[x, y]$ . Složitost všech operací včetně této by měla být  $O(\text{hloubka})$ .