12. cvičení

Knihovna 2. Podobně jako v předchozím cvičení chceme navrhnout knihovnu, jež pojme dané knihy. Tentokrát ovšem máme zadanou maximální výšku knihovny a chceme najít minimální možnou šířku. Pokud vám to pomůže, předpokládejte, že všechny knihy mají jednotkovou šířku.

Depeše. Dešifrovali jsme tajnou depeši, ale chybí v ní mezery. Známe však slovník všech slov, která se v depeši mohou vyskytnout. Chceme tedy rozdělit depeši na co nejméně slov ze slovníku.

Edit – prostor. Algoritmus EDIT2 zabere $\Theta(nm)$ buněk paměti na uložení tabulky T. Ukažte, jak spotřebu paměti snížit na $\Theta(n+m)$.

Podobné řetězce. Na první pohled se zdá, že čím podobnější řetězce dostaneme, tím by mělo být jednodušší zjistit jejich editační vzdálenost. Náš algoritmus ovšem pokaždé vyplňuje celou tabulku. Ukažte, jak ho zrychlit, aby počítal v čase $\mathcal{O}((n+m)(L(x,y)+1))$.

Váhy 1. Jsou dány rovnoramenné váhy a 3 různě těžké kuličky. Ukažte, že je není možné uspořádat dle hmotnosti na méně než 3 vážení. Co se změní, pokud je cílem pouze najít nejtěžší kuličku?

Váhy 2. Jsou dány rovnoramenné váhy a 12 kuliček, z nichž právě jedna je těžší než ostatní. Na misku lze dát i více kuliček naráz. Navrhněte, jak na 3 vážení najít těžší kuličku. Dokažte, že na 2 vážení to není možné.

Váhy 3. Jsou dány rovnoramenné váhy a 12 kuliček, z nichž právě jedna je jiná než ostatní, nevíme však zda je lehčí nebo těžší. Na misku lze dát i více kuliček naráz. Navrhněte, jak na 3 vážení najít tuto jinou kuličku a zjistit, jestli je lehčí nebo těžší. Dokažte, že na 2 vážení to nejde.

Váhy 4. Stejná úloha jako předchozí, avšak s 13 kuličkami. Dokažte, že stále stačí 3 vážení, pokud slevíme z požadavku zjistit, zda je odlišná kulička lehčí nebo těžší.

Váhy obecně. Řešte cvičení Váhy 1 obecně pro n kuliček a navrhněte algoritmus používající co nejmenší počet vážení. Dokažte, že tento počet je optimální.

Průměrná složitost. Dokažte, že $\Omega(\log n)$ resp. $\Omega(n \log n)$ porovnání je potřeba nejen v nejhorším případě, ale i v průměru přes všechny možné vstupy. V případě vyhledávání průměrujeme přes všechna možná hledaná x, u třídění přes všechny permutace.

Matice. Mějme matici A tvaru $n \times n$, v níž jsou uložena celá čísla a navíc každý řádek i sloupec tvoří rostoucí posloupnost. Jak najít i, j takové, že $A_{i,j} = i + j$? Pokud existuje více řešení, stačí vypsat jedno. Čas na načtení matice do paměti nepočítáme.
*Hint:*Podívejte se na levý dolní prvek matice. Co plyne z $A_{i,j} < i + j$ a co z $A_{i,j} > i + j$?

Úsporný medián. Nalezněte k-tý nejmenší z n prvků, máte-li k dispozici pouze paměť' asymptoticky menší než k. Pokuste se dosáhnout lepší časové složitosti než $\Theta(kN)$.

Náhodné stromy. Uvažujme binární vyhledávací strom, který vznikl postupným vkládáním hodnot $1, \ldots, n$ v náhodném pořadí, bez jakéhokoliv vyvažování. Dokažte, že střední hodnota průměrné hloubky vrcholu je $\mathcal{O}(\log n)$. (Pro jistotu: průměr je obyčejný aritmetický, nijak v něm nefiguruje náhoda; z těchto průměrů pak počítáme střední hodnotu přes všechny možné průběhy algoritmu.) Napovíme, že náhodné stromy souvisí s možnými průběhy Quicksortu.