

# Programování pro matematiky

## 5. cvičení - Datové struktury

Peter Kovács

---

Doporučuje si promyslet řešení každého z úkolů, pro vaše vlastní ujasnění učiva. Navíc pokud některé z úvah sepišete můžete získat body

Každý úkol obsahuje za názvem maximální počet bodů, které lze za úkol získat. Vaším úkolem je si vybrat úkoly, které chcete řešit. Klidně všechny. Úkoly, které jste si vybrali sepište do jednoho souboru a odvezďte do recodexu. Za celý úkol můžete získat maximálně **2 body**. Po odvezdání ohodnotím každou odevzdanou úlohu. Body sečtu a přidělím vám  $\max(\text{body}, 2)$ . Úkoly řešte sami. Pokud spolupracujete v skupině dostanete své body podělené počtem lidí v skupině.

### Uzavorkovani - implementace (1 bod):

Do kódu z cvičení doimplementujte kontrolu uzávorkování do funkce `skontroluj_uzavorkovani(zavorky)`. Na vstupu dostanete list s hodnotami. Kde kladné číslo  $i$  značí otevření závorky typu  $i$  a  $-i$  značí uzavření závorky typu  $i$ . Při řešení můžete použít předimplementované kusy kódu v zdrojáku. V zdrojáku jsou za funkcí testy, které odskoušejí zda implementace funguje. Pokud se rozhodnete řešit tento úkol odevzdejte zdroják do recodexu spolu s řešením jiných úloh. Kód se nevyhodnotí automaticky a budu ho bodovat ručně.

### Seřazení v grafu (1 bod):

Popište algoritmus, který by dostal na vstup souvislý graf a jeden vrchol z grafu a vypsal všechny vrcholy grafu seřazené dle vzdálenosti od zadaného vrcholu. V každém vrcholu grafu víme zavolat funkci `sousedí(v)`, která vrátí list, všech sousedů vrcholu  $v$  v grafu. Graf není ohodnocen a teda délka cesty mezi vrcholy je rovna počtu hran. Pokud jsou 2 vrcholy stejně vzdálené je jedno v jakém pořadí jich vypíšeme. Dále popište jak by sme postupovali pokud by bol graf ohodnocený a zároveň acyklický. Algoritmy popište slovně nebo v pseudokódu.

### Poloha prvků v haldě (0 bod) - není za body ale určité doporučuje si to přemyslet:

1. Kde všude se může nacházet druhý největší prvek?
2. Kde všude se může nacházet třetí největší prvek?
3. Kde všude se může nacházet nejmenší prvek?
4. Kde všude se může nacházet druhý nejmenší prvek?

### (ne)porušení haldové podmínky (1 bod):

Podívejme se na co se stane po odstránení prvku z haldy. Navrch vyhodíme poslední prvek z haldy a ten bude bublat dolu (BubbleDown). Na cvičení sme si ukázali, že prvek může porušovat haldovou podmínku při cestě nadol. My tuto podmínku upravujeme tak, že pokud je nějaký ze synů menší jako otec, tak vyměníme otce a menšího z dvou synů. Tím sme urovnali haldovou podmínku pro náš prvek a možná porušili haldovou podmínku o řověn níže. Ukažte, že syn, který si s otcem vyměnil místo neporušuje haldovou podmínku.

### Prioritní fronta (1 bod):

Prioritní fronta se někdy definuje tak, že prvky se stejnou prioritou vrací v pořadí, v jakém byly do fronty vloženy. Ukažte, jak takovou frontu realizovat pomocí haldy. Dosáhněte časové složitosti  $O(\log n)$  na operaci.