2. cvičení z ADS1

24. 2. 2022

Příklad 1.

Mějme program v RAMu, kde dopředu na základě vstupu velikosti n umíme v čase $\mathcal{O}(f(n))$ a v konstantním prostoru určit, že všechna používaná čísla budou v rozsahu $0, \ldots, m$. Na základě toho ukažte, jak tento program převedete na jiný program, který kromě vstupu využije $\mathcal{O}(1)$ buněk.

Zkuste nejprve vymyslet, jak zakódovat buňky, a potom pro jednotlivé instrukce ukázat, jak pracovat s tímto kódováním.

Ukažte taky, jak toto kódování změnit, pokud mohou být čísla libovolně velká, tedy neznáme dopředu rozsah.

Příklad 2.

Uvažme model časové složitosti, kde máme neomezená čísla a jednotkovou cenu všech instrukcí.

Jak se asymptoticky změní časová složitost programu, jestliže původně byla $\mathcal{O}(g(n))$, pokud použijeme předchozí převod?

Příklad 3.

Musí platit, že prostorová složitost RAMu je shora omezená časovou?

Příklad 4.

Jak se změní odpověď na předchozí otázku, pokud změníme definici prostorové složitosti jako počet různých buněk:

- do kterých jsme někdy něco zapsali
- ze kterých jsme někdy něco přečetli
- ke kterým jsme někdy přistoupili

Příklad 5.

Víme, že DFS je rekurzivní, tudíž využívá zásobník na ukládání vrcholů. BFS naopak na ukládání vrcholů využívá frontu. Nabízelo by se tedy zkusit DFS naimplementovat tak, že vezmeme implementaci BFS, ve které nahradíme frontu zásobníkem.

Jaké vlastnosti bude mít tento algoritmus?

Příklad 6.

Zkuste DFS naimplementovat bez použití rekurze tak, aby uměl i klasifikovat hrany. Pokud lze využít implementaci z předchozího příkladu, odůvodněte proč.

Definice. Vrchol $v \in V(g)$ je *artikulace*, jestliže $G - v$ má více komponent než G.

Příklad 7.

Jaké vlastnosti musí pro $v \in V(G)$ platit, aby v byla artikulace? Zkuste využít klasifikaci hran. Pomocí této vlastnosti pak upravte DFS tak, aby umělo hledat artikulace G.