

Amortizovaná složitost

Amortizovaná časová složitost označuje časovou složitost algoritmu v sekvenci nejhorších možných vstupních dat. Na rozdíl od průměrné složitosti nevyužívá pravděpodobnosti a je proto zaručená.

Amortizovaná vs. asymptotická složitost

Asymptotická složitost je deklarována na základě nejhorší (nejlepší) možné instance běhu algoritmu, což ale není vždy vypovídající, protože i nejhorší sekvence případů může mít výrazně lepší průběh, než by asymptotická složitost napovídala. Tento zdánlivý paradox je zapříčiněn tím, že operace s vysokou složitostí změní datovou strukturu tak, že takto špatný případ nenastane po nějakou delší dobu - tím se složitá operace amortizuje.

Příklad

Mějme *dynamické pole* (*ArrayList*), které se natahuje klasickým způsobem – je-li jeho kapacita vyčerpána, naalokuje se dvojnásobné pole a to původní se do něj překopíruje.

Asymptotická složitost operace vkládání je očividně $O(n)$, protože v nejhorším případě dojde k realokaci pole a kopírování všech prvků. Amortizovaná složitost, jak si nyní ukážeme, je pouze $O(1)$.

Účetní metoda

Pro výpočet použijeme tzv. *účetní metodu*, při které je každé operaci přiřazena amortizovaná a skutečná cena. Pokud je amortizovaná cena vyšší než skutečná, tak rozdíl vložíme na společný účet, je-li skutečná cena vyšší, pak rozdíl z účtu vybereme. V každém okamžiku musí být zůstatek na účtu nezáporný.

Vkládáme prvky do pole a stanovme této operaci amortizovanou cenu $(2k + 1)$ žetonů, kde k je nějaká konstanta. První žeton vždy spotřebujeme na samotnou operaci insert (skutečná cena). Zbylé žeton uchovíme na společném účtu. V okamžiku, kdy se původní pole naplní, zaalokujeme nové, překopírujeme prvky a smažeme staré pole (složitost této operace na jeden prvek stanovme jako k , pro n prvků $k \cdot n$). Protože jsme od minulé alokace pole přidali přesně $n/2$ prvků, máme na účtu $k \cdot n$ žetonů, což přesně zmíněné cena realokace pole. Amortizovaná složitost algoritmu je proto $O(2k + 1) = O(1)$.

Tento závěr mj. odpovídá na otázku, proč může být dynamické pole výkonnější než *spojový seznam*, ač vkládání jeho prvků vyžaduje občasnou realokaci.

Použití

Amortizovaná složitost dává přesnější popis dlouhodobého chování systému než asymptotická složitost. Její použití se však omezuje pouze na situace, kdy nemusíme garantovat okamžitou odezvu systému, protože vždy může nastat ona jednotlivá nejhorší situace.

– Pavel Mička

Zakladatel a administrátor stránek Algoritmy.net