

Domácí úkol 1 - Reprezentace množiny

Termín odevzdání: Nejpozději na cvičení 4. dubna

Vaším úkolem je navrhnout způsob, jak v počítači efektivně reprezentovat množinu přirozených čísel.

Při programování občas potřebujeme pracovat s množinami prvků. Například při hledání cesty v grafu potřebujeme udržovat množinu vrcholů, které jsme už navštívili, abychom je zbytečně neprohledávali víckrát. K tomu je třeba mít vhodnou datovou reprezentaci množiny, která by umožňovala provádět následující operace: přidání prvku do množiny, odebrání prvku z množiny a testování, jestli daný prvek v množině je, nebo ne. Pro jednoduchost se zaměříme pouze na množiny přirozených čísel, a pouze konečné a shora omezené.

Formálně: je dané číslo M - horní mez na velikost prvků v množině. Vaším úkolem je navrhnout program, který bude reprezentovat množinu $S \subset \{1, 2, \dots, M\}$ (na začátku prázdnou), a který bude implementovat tyto procedury / funkce:

- procedure *pridej*(prvek : integer)
přidá nový prvek do množiny, čili $S_{new} = S \cup \{prvek\}$ (S je množina, kterou program reprezentoval před zavoláním procedury a S_{new} je množina, kterou reprezentuje po skončení procedury. Může se stát, že prvek už v množině S byl obsažený.)
- procedure *smaz*(prvek : integer)
smaže zadaný prvek z množiny, čili $S_{new} = S \setminus \{prvek\}$ (S je množina, kterou program reprezentoval před zavoláním procedury a S_{new} je množina, kterou reprezentuje po skončení procedury. Může se stát, že prvek v množině S nebyl obsažený.)
- function *obsahuje*(prvek : integer) : boolean
vrátí *TRUE* pokud $prvek \in S$ jinak vrátí *FALSE*. (S je množina, kterou program reprezentoval před zavoláním funkce. Tímto voláním se S nezmění.)

Jaké proměnné a datové struktury v programu využijete, je na vás, ale je potřeba, aby procedury / funkce fungovaly korektně. (Nesmíte samozřejmě používat Pascalovský datový typ *set*.)

Nemusíte psát přímo korektní zdrojový kód, stačí když slovně popíšete, jak a kde budou uložena data a jak budou fungovat procedury / funkce. (Zdrojový kód bez tohoto slovního popisu **nebudu** považovat za správné řešení úkolu.) Zaměřte se spíše na analýzu, zhodnoťte možné způsoby řešení, popište jejich výhody a nevýhody a vyberte nejvhodnější variantu. Součástí řešení by mělo být také určení **časové a paměťové složitosti** navržených algoritmů.

Poznámka ke složitosti Složitost algoritmu je funkce, která popisuje závislost spotřebovaného času případně paměti na velikosti vstupu. Jak budete měřit velikost vstupu (teda jaké budou argumenty funkce) je na vás. Složitost může záviset např. na čísle M , $|S|$, $\max\{S\}$, na počtu zavolání procedury *pridej* a podobně (podle toho, jaký způsob řešení zvolíte). Můžete předpokládat, že velikost reprezentované množiny bude po celou dobu běhu programu výrazně menší než M ($|S| \ll M$)

Bodové ohodnocení Nejvýše 50 bodů získáte pokud

- navrhnete 2 způsoby jak reprezentovat množinu
- popíšete jak budou fungovat procedury / funkce deklarované výše. Pokud budete používat nějaké netriviální techniky, měli byste dokázat, že procedury / funkce fungují správně (mají požadované vlastnosti).
- spočítáte časovou a paměťovou složitost v nejlepším a v nejhorším případě. (čím menší bude, tím líp.)

Dalších max 50 bonusových bodů můžete získat za tyto rozšiřující úkoly

- navrhnete více než 2 způsoby
- zkusíte spočítat složitost v průměrném případě (průměrný případ definujte sami nějak rozumně - např. tak, že všechna čísla z $\{1, 2, \dots, M\}$ budou mít v průměru stejnou šanci, že budou argumentem procedur *pridej* a *odeber* a funkce *obsahuje*).
- vyřešíte tyto speciální případy: (pro každý z nich buď navrhnete nový způsob reprezentace, nebo použijte předchozí a spočítejte znovu časovou a paměťovou složitost)
 - nebude se používat procedura *odeber* - čili budou se pouze přidávat prvky a testovat zda je prvek v množině.
 - množina, kterou má program reprezentovat, je předem známá a nemění se. Je daná množina S např. jako posloupnost prvků na vstupu. Program ji přečte a uloží. Potom už se bude volat pouze funkci *obsahuje* (*pridej* ani *odeber* se volat nebudou).

V případě nejasností v zadání mi pošlete e-mail nebo se zeptejte na cvičení. Řešení můžete posílat buď e-mailem nebo odevzdat na papíře na cvičení.