Zkouška z Algoritmů a Datových Struktur I (24. 5. 2007)

Příklad A1 (5 bodů). Definujte (a, b)-strom a dokažte, že má hloubku $\mathcal{O}(\log n)$.

Příklad A2 (5 bodů). Popište algoritmus pro hledání k-tého nejmenšího prvku v lineárním čase. Dokažte, že funguje, a rozeberte časovou a paměťovou složitost.

Příklad B1 (5 bodů). Je dán neorientovaný graf s hranami ohodnocenými přirozenými čísly $1, 2, \ldots, K$. Navrhněte co nejefektivnější algoritmus pro nalezení minimální kostry takového grafu.

Příklad B2 (5 bodů). Je dána posloupnost reálných čísel $a_1, a_2, \ldots a_n$. Navrhněte co nejefektivnější algoritmus, který spočítá, kolik má tato posloupnost inverzí, čili kolik existuje dvojic (i, j) takových, že $i < j$ a $a_i > a_j$.

Příklad C (5 bodů). Fibonacciho číselná soustava funguje tak, že čísla zapisujeme pomocí nul a jedniček a zápis $\langle c_n \ldots c_2 \rangle$ odpovídá číslu $\sum_i c_i F_i$, kde F_i je i-té Fibonacciho číslo ($F_0 = 0$, $F_1 = 1$, $F_{n+2} = F_n + F_{n+1}$). Zápisu, v němž se nevyskytují dvě jedničky za sebou, budeme říkat hezký. Platí, že každé přirozené číslo má právě jeden hezký zápis. Vymyslete algoritmus, který v lepším než kvadratickém čase sečte dvě čísla zadaná hezkými Fibonacciho zápisy a vydá hezký zápis výsledku.

Poznámky:

Příklady jsou tří druhů: teoretické $\mathbf{A}i$, u kterých byste měli vše precizně formulovat a zdůvodnit, dále praktické $\mathbf{B}i$, kde se můžete odkazovat na algoritmy a věty z přednášky, aniž byste je museli odvozovat, a konečně nepovinný příklad **C**, jenž slouží jako lahůdka pro ty, kdo budou s písemkou dříve hotovi.

Ke každému algoritmu neodmyslitelně patří rozbor jeho správnosti (není-li zjevná) a časové a paměťové složitosti.

Při zkoušce je zapovězeno používat zápisky, kalkulačky, mobily, své kolegy, jakož i jiné pomůcky. Společně vyřešené úlohy budou obodovány taktéž společně.

Hodně štěstí!