

Programování II (NPRM045) – Cvičení 10

Obsah cvičení

- odstraňování prvků z BVS
- složitější rekurzivní operace na stromě

Příklady

1. Písemka: Implementujte proceduru provádějící pravou nebo levou rotaci (dle své politické orientace).
2. Implementujte funkci `odeberNejpravejsi` (`var koren: UkUzel`): `Integer`, která odebere ze stromu prvek s největší hodnotou (tj. prvek nacházející se co nejvíce vpravo) a jeho hodnotu vrátí. Parametr `koren` je nenilový ukazatel na kořen stromu, z něž chceme odebrat prvek.
3. Implementujte proceduru `odstranKoren` (`var koren: UkUzel`), která odstraní kořen stromu. Parametr `koren` je nenilový ukazatel na kořen stromu.
4. Implementujte proceduru, která z binárního vyhledávacího stromu vypustí prvek se zadanou hodnotou.
5. Rozmyslete si, jak byste řešili tuto úlohu: Napište proceduru `procedure rozděl` (`var koren, s1, s2: UkUzel; kriterium: integer`), která strom s kořenem `koren` rozdělí na strom obsahující všechny prvky původního stromu, které jsou menší nebo rovny `kriterium` (jeho kořenem bude `s1`) a na strom obsahující všechny prvky původního stromu, které jsou větší než `kriterium` (jeho kořenem bude `s2`). V ideálním případě by zdrojový kód procedury neměl být výrazně delší než asi 15 řádek. Náповěda: rekurze je mocná čarodějka.

Domácí úkol

1. Napište rekurzivní proceduru, která zjistí výšku binárního vyhledávacího stromu.
2. Implementujte proceduru dělící strom na dva stromy dle zadaného kritéria, kterou jsme si rozmýšleli na cvičení.
3. Implementujte proceduru, která vypustí z BVS všechny prvky, které nepatří do zadaného intervalu (tuto úlohu si určitě zkuste).
4. Implementujte proceduru, která postaví vyvážený BVS z čísel ze zadaného intervalu (návod: do kořene dát číslo z prostředku intervalu, pak rekurzivně postavit levý a pravý podstrom).
5. Implementujte proceduru, která postaví vyvážený BVS z uspořádaného lineárního spojového seznamu (návod: zkuste využít proceduru `split`, kterou jsme programovali na jednom z předchozích cvičení, využijte rekurzi).
6. Analyzujte následující úlohu: Volí se prezident. Voličů jsou řádově stovky milionů (jedná se tedy o velkou zemi). Země je rozdělena na desítky volebních krajů, v každém kraji jsou řádově stovky volebních okrsků, v okrscích tisíce volebních místností. Voliči i kandidáti jsou identifikováni něčím jako rodné číslo (12timístné číslo). Kandidátů je nejvýše 100.

Volič na hlasovacím lístku uvede úplné pořadí kandidátů (tedy seřadí všechny kandidáty podle svých preferencí). Kromě toho může některé kandidáty vetovat, nejvýše však může vetovat pět kandidátů. Vyhodnocení probíhá následovně: Pokud je v některém kraji kandidát vetován více než 10 % platných hlasů, je vetován krajem. Kandidát vetovaný více než dvěma kraji automaticky vypadne. V každém kroku se provede toto: Spočítá se počet prvních míst na platných volebních lístcích pro každého kandidáta. Je-li někdo uveden na prvním místě na více než 50 % platných lístků, stává se prezidentem. Pokud ne, pro každého kandidáta se spočítá koeficient – součet přes všechny hlasovací lístky pořadí, na kterém je na hlasovacím lístku uveden. Kandidáti (může jich být i více, pokud mají stejný koeficient) s maximálním koeficientem vypadnou a stejný krok (zjištění, zda je na více než 50 % platných lístků, spočítání koeficientu,...) opakujeme, dokud z algoritmu nevypadne jméno prezidenta.

Navrhněte způsob komunikace mezi volebními místnostmi, volebními okrsky, kraji a ústřední volební komisí. Cílem je minimalizovat objem přenesených dat.