Név: ..... *Neptun kód:* .....

Algoritmusok és adatszerkezetek I. vizsga, 2016.01.26.

1. **(1.a)** Hozzuk postfix formára az „`5+8*(4/(9-7))-6*2/3`” kifejezést, és **(1.b)** értékeljük ki az „`58497-/*+62*3/-`” lengyel formát, a gyakorlatról ismert algoritmusokat szemléltetve! (Feltesszük, hogy mindegyik szám egyjegyű, és mindegyik operátornak két operandusa van.) A vermet minden egyes kipakolás után újra kell lerajzolni.

Mekkora a lengyel forma kiértékelő algoritmus műveletigénye, ha mindegyik operandus egyjegyű szám, és mindegyik operátornak két operandusa van? Miért? (20p)

2. Milyen magas egy B+ fa? Milyen kapcsolatban áll ez a keresés, a beszúrás és a törlés műveletigényével? Adott az
{ | (1 2) 3 (4 5) | 6 | (7 8) 9 (9 10) 11 (11 12 13) 14 (15 16) | }
negyedfokú B+ fa. Rajzoljuk le a fát! Szemléltessük az előadásról ismert algoritmus szerint a 11, a 15 és a 4 törlését, **mindhárom esetben az eredeti fára**! (20p)

3. Az $A[1..m]$ tömb első n elemében egy bináris kupacot tárolunk
$(m > 0 \wedge 0 \leq n \leq m)$.

Írjuk meg a **maxKivesz**$(A[1..m], n, x)$ függvényt, ami kiveszi a kupacból a legnagyobb elemét, és x-be teszi! Akkor ad vissza **igaz** logikai értéket, ha a művelet sikeres volt. Próbáljunk meg minél hatékonyabb algoritmust írni! Mekkora a műveletigénye? Miért? (20p)

4. Az L_1 pointer egy nemüres, fejelemes láncolt lista fejelemére mutat, az L_2 pointer pedig egy egyszerű láncolt lista elejére ($L_2 = \emptyset$ is lehet).

Írjuk meg a **maxÁtfűz**(L_1, L_2) eljárást, ami az L_1 lista legnagyobb kulcsú elemét *átfűzi* az L_2 lista elejére! A program az L_1 listán csak egyszer menjen végig! $T(n) \in \Theta(n)$, ahol n az L_1 lista hossza. A listaelemeknek a *kulcs* és a *mut* (mutató) mezőkön kívül más részei is lehetnek, de ezeket nem ismerjük. (A listaelemeket tehát nem tudjuk lemásolni.) (20p)

5. A bináris fákat ismertnek feltételezve, mondjuk ki az AVL fa meghatározásához szükséges definíciókat! Az AVL fa mérete és magassága között milyen összefüggést ismer? Mi ennek a jelentősége az AVL fa műveletei szempontjából? Melyek ezek a műveletek?

Adjuk meg az előadásról ismert **AVLbeszúr**(t, k, d) rekurzív eljárás struktogramját, ami a t AVL fába beszúrja a k kulcsot, a d logikai típusú paraméterben pedig visszadja, hogy a művelet hatására nőtt-e a fa magassága! A **balRészfaNőtt**(t, d) és a **jobbRészfaNőtt**(t, d) segédeljárásokat nem kell részletezni. (20p)