

Név: Neptun kód:

Algoritmusok és adatszerkezetek II. vizsga, 2016. 06. 14.

1. Mutassa be a *számjegypozíciós (Radix) rendezés* működését a következő, négyes számrendszerbeli számok tömbjén: $\langle 21; 01; 20; 02; 31; 20 \rangle$! Az egyes menetekben *leszámláló rendezést* alkalmazzon! Mekkora a *Radix rendező* algoritmus műveletigénye? A *leszámláló rendezés* – mint segédprogram – mely tulajdonságaira épül a *Radix rendezés*? (20p)

2. A $T[0..m-1]$ hasító tábla rései kétirányú, nemciklikus, fejelem nélküli, rendezetlen láncolt listák pointerai. Adott a $k \bmod m$ hasító függvény. A kulcsütközéseket láncolással oldjuk fel. Mindegyik kulcs csak egyszer szerepelhet T -ben. **(2.a)** Írjuk meg az $\text{ins}(T[], m, k, a):0..2$ értékű függvényt, ami beszúrja a hasító táblába a (k, a) kulcs-adat párt! Ha a táblában már volt k kulcsú elem, a beszúrás megghiúsul, és a 2 hibakódot adja vissza. Különben, ha nem tudja már a szükséges listaelemet allokálni, az 1 hibakódot adja vissza. (Feltesszük, hogy a **new** művelet, ha sikertelen, akkor \ominus ponttert ad vissza.) Az $\text{ins}()$ művelet akkor ad vissza 0 kódot, ha sikeres volt a beszúrás. Ilyenkor az új listaelemet a megfelelő lista elejére szúrja be. **(2.b)** Mi a kitöltöttségi hányados? Milyen aszimptotikus becslést tud adni a fenti művelet minimális, átlagos és maximális futási idejére? Miért? (20p)

3. Mit számol ki a *Szélességi gráfkeresés*? Szemléltesse a következő irányítatlan gráfon, az **a** csúcsból indítva!¹. $a - b$; $d - b - c$; $d - c - e$. $d - e$.

Adja meg az algoritmus absztrakt struktogramját! Mely csúcsokba, és a végrehajtás során mikor találjuk meg az optimális utat? Mit tud a *Szélességi gráfkeresés* műveletigényéről? (Indokolja is az állítást!) (20p)

4. Mit jelent egy gráf tranzitív lezártja, amit *Warshall* algoritmusa számol ki? Tegyük fel, hogy a gráfnak n csúcsa van! Mi a $\langle T^{(k)} : k \in 0..n \rangle$ mátrix sorozat szerepe, definíciója és rekurzív képlete? Adja meg az algoritmus struktogramját! Mekkora a műveletigénye n csúcsú gráf esetén? Miért elegendő egyetlen T mátrix a programban? Mutassa be az algoritmus működését az $1 \rightarrow 2$. $2 \rightarrow 3$. $3 \rightarrow 4$. $4 \rightarrow 2$. irányított gráfon! (20p)

5. Szemléltessük a *Huffman kódolás* működését az *ÁBRÁBANÁBRA* szövegen! Adjuk meg a kódfát és a szótárat! Mekkora a *Huffman kódolással* tömörített kód hossza? Mekkora lenne a tömörített kód *fix hosszú* karakterkódok esetén? Hogyan dekódolható egy *Huffman kóddal* tömörített szöveg? Milyen értelemben optimális a *Huffman kód*? Azt jelenti-e ez, hogy a *Huffman kódolás* a lehető legjobb tömörítés? Miért? (20p)

¹ $u - v_1; \dots v_k$. azt jelenti, hogy az *irányítatlan gráfban* az u csúcs u -nál nagyobb indexű szomszédai $v_1, \dots v_k$. (Ezzel a jelöléssel a gráf minden élét csak egyszer tüntetjük fel.)