

Név: ..... Neptun kód: .....

## Algoritmusok és adatszerkezetek II. vizsga, 2016. 05. 24.

1. Mutassuk be a számjegypozíciós („Radix”) rendezés működését a  $\langle 11; 20; 10; 23; 21; 30 \rangle$  négyes számrendszerbeli számok listáján! Az egyes menetekben a megfelelő számjegy szerinti edényrendezést alkalmazzuk! Mekkora a Radix rendező algoritmus műveletigénye? A felhasznált edényrendezés – mint segédprogram – mely tulajdonságaira épül a Radix rendezés? (15p)

2. A  $T[0..m-1]$  hasító tábla rései egyirányú, nemciklikus, fejelem nélküli, rendezetlen láncolt listák pointerei. Adott a  $k \bmod m$  tördelő függvény. A kulcsütközést 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! Akkor és csak akkor ad vissza 0 értéket, ha sikeres volt a beszúrás. Ilyenkor az új listaelemet a megfelelő lista elejére szúrja be. Ha a táblában már volt  $k$  kulcsú elem, a beszúrás meghiúsul, 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 sikertelenség esetén  $\odot$  pointert ad vissza. **(2.b)** Írjuk meg a  $\text{search}(T[], m, k)$  függvényt, ami visszaadja a  $T$ -beli,  $k$  kulcsú elem címét, vagy a  $\odot$  pointert, ha ilyen nincs! **(2.c)** Mi a kitöltöttségi hányados? Milyen becslést tudunk adni a fenti műveletek aszimptotikus futási idejére? (25p)

3. Milyen feladatot old meg a Knuth-Morris-Pratt (KMP) algoritmus? Szemléltessük a KMP algoritmus (a)  $\text{init}(\text{next} \dots)$  eljárásának működését az *ABACABA* mintán és (b) e mintát illesztő eljárását az *ABABACABACABABACABABA* szövegen! Mekkora az egyes eljárások műveletigénye? Mi KMP algoritmus előnye, illetve hátránya a Quick-search mintaillesztő algoritmussal összehasonlítva? (20p)

4. Mit számol ki a Floyd-Warshall algoritmus? Mekkora a műveletigénye  $n$  csúcsú gráf esetén? Miért? Szemléltessük a működését az alábbi irányítatlan gráfon a  $(D^{(0)}, \Pi^{(0)}), \dots, (D^{(4)}, \Pi^{(4)})$  mátrix párok megadásával<sup>1</sup>. Melyik az alábbi gráf legkisebb részgráfja, ami az összes optimális utat tartalmazza?

$1 - 2, 3; 3, 1; 4, 4. \quad 2 - 4, 0. \quad 3 - 4, 1. \quad 4.$  (20p)

5. Mit számol ki a Dijkstra algoritmus? Adja meg a struktogramját! Mit értünk a gráfok élsúlyozott szomszédossági listás ábrázolása alatt? Mekkora az algoritmus futási ideje az előbbi gráfrepresentáció és a prioritásos sor bináris kupaccal való megvalósítása esetén? Miért? Milyen állítás igaz, amikor egy tetszőleges csúcsot kiválasztunk kiterjesztésre? Miért? (20p)

<sup>1</sup> $u - v_1, w_1; \dots v_k, w_k$ . azt jelenti, hogy a gráfban az  $u$  csúcs  $u$ -nál nagyobb indexű szomszédai  $v_1, \dots v_k$ , és a megfelelő irányítatlan élek súlyai sorban  $w_1, \dots w_k$ .