

Név: Neptun kód:

Algoritmusok és adatszerkezetek II. vizsga, 2016. 06. 07.

1. Adott az L egyszerű láncolt lista, aminek $n \geq 0$ eleme van. Minden elemének kulcsa a $[0; 1)$ intervallumon egyenletes eloszlás szerint választott számérték. (A listaelemeknek más mezői is lehetnek, amiket nem ismerünk.) Írja meg a $\text{bucketSort}(L, n)$ utasítással meghívható egyszerű edényrendezés struktogramját, $AT(n) \in \Theta(n)$, $MT(n) \in \Theta(n \lg n)$ műveletigénnyel és $M(n) \in O(n)$ tárigénnyel! Segédrendezésként felhasználható a megfelelő, múlt félévben tanult, egyszerű láncolt listákat rendező eljárás. Ezt nem kell megírni, a kód többi részét viszont teljes részletességgel kérjük. (20p)

2. A $T[0..m-1]$ hasító táblában a kulcsütközést nyílt címezéssel oldjuk fel. Mit értünk *kitöltöttségi hányados*, *próbasorozat* és *egyenletes hasítás* alatt? Mekkora egy sikertelen keresés várható hossza 80%-os kitöltöttség esetén, ha nincs törölt rés? Egy sikeres keresésé ennél több vagy kevesebb? Miért?

Legyen most $m = 11$, $h'(k) = k \bmod m$, és alkalmazzunk lineáris próbát! Az alábbi műveletek mindegyikére adja meg a *próbasorozatot* $\langle \dots \rangle$ alakban! Szemléltesse a hasító tábla változásait is! Szűrjük be a táblába sorban a következő kulcsokat: 10; 22; 31; 4; 15; 28; 16; 26; 62; ezután töröljük a 16-ot, majd próbáljuk megkeresni a 27-et és a 62-t, végül pedig szűrjük be a 27-et!

3. Milyen feladatot old meg a *Knuth-Morris-Pratt (KMP)* algoritmus? Defináljuk a *next* függvényt, majd adjuk meg a *BABABAB* mintán! Szemléltessük *KMP* algoritmussal e minta előfordulásainak keresését a *BABBABABABABBABABABAAB* szövegben! Mi a *next* függvény szerepe a keresés során? Mi *KMP* algoritmus előnye, illetve hátránya a *Quick-search* mintaillesztő algoritmussal összehasonlítva? (20p)

4. Mit számol ki a *Floyd-Warshall* algoritmus? Tegyük fel, hogy a gráfnak n csúcsa van! Mi a $\langle (D^{(k)}, \Pi^{(k)}) : k \in 0..n \rangle$ mátrix-pár sorozat szerepe, definíciója és rekurzív képlete? Adja meg az algoritmus struktogramját! Mekkora a műveletigénye n csúcsú gráf esetén? Miért elegendő egyetlen (D, Π) mátrix-pár a programban? (20p)

5. Mit számol ki a *Prim* algoritmus? Szemléltesse a működését az alábbi gráfon, a **d** csúcsból indítva¹.

a – b, 0; d, 1. b – c, 5; d, 2; e, 3. c – e, 2. d – e, 2.

Mondja ki a biztonságos élekről és a minimális feszítőfákról szóló tételt! Definálja a tételben szereplő *vágás* és *könnyű él* fogalmakat! Hogyan következik a *Prim* algoritmus helyessége ebből a tételből? (20p)

¹ $u - v_1, w_1; \dots v_k, w_k$. 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 $w_1, \dots w_k$. (Ezzel a jelöléssel minden élet csak egyszer tüntetünk fel.)