Név: .... *Neptun kód:* ....

7. Algoritmusok és adatszerkezetek II. vizsga, 2016. 06. 28.

1. Mutassuk be a számjegypozíciós („Radix") rendezés működését a $\langle 31; 20; 11; 23; 21; 10\rangle$ négyes számrendszerbeli számok listáján! Az egyes menetekben a megfelelő számjegy szerinti edényrendezést alkalmazzuk! Mekkora a felhasznált – adott számjegy szerinti edényrendezés – és a ráépülő Radix rendezés műveletigénye? A felhasznált edényrendezés – mint segédprogram – mely tulajdonságára épül a Radix rendezés? (20p)

2. $G[1..n]$ egy irányított, élsúlyozott, egyszerű gráf szomszédossági éllistás ábrázolása. A $G[i]$ éllisták egyszerű láncolt listák. Írjuk meg az FWinit$(G[1..n], D[1..n, 1..n], \Pi[1..n, 1..n])$ eljárást, ami $G[1..n]$ szerint a Floyd-Warshall algoritmus $(D^{(0)}, \Pi^{(0)})$ mátrix-párjának megfelelően inicializálja az eljárás D és Π paramétereit! $T(n) \in \Theta(n^2)$. (20p)

3. Mit értünk egy irányított gráf csúcsainak topologikus rendezése alatt? Mondja ki és bizonyítsa be az ezzel kapcsolatos tételt! Mutassa be az alábbi gráfcsúcsai topologikus rendezésének a gráf mélységi bejárására épülő algoritmusát! (Az algoritmus szemléltetése során a nemdeterminisztikus esetekben mindig az alfabetikusan kisebb indexű csúcsot részesítse előnyben!) Módosítsa egyetlen él behúzásával úgy a gráfot, hogy ne legyen topologikus rendezése! A módosított gráfnak miért nincs topologikus rendezése? Mikor derül ez ki a fent szemléltetett algoritmus végrehajtása során? Mit tud a mélységi bejárásra épülő topologikus rendezés műveletigényéről? (Indokolja is az állítást!) a $\to$ b; d. b $\to$ c; d. c $\to$ e. d $\to$ e. f $\to$ c; e. (20p)

4. Írja le röviden, szóban vagy struktogrammal, azt a tanult algoritmust, mellyel irányított, súlyozott, *körmentes* gráfokra a leghatékonyabb módon határozhatjuk meg a start csúcsból a többi csúcsba vezető legrövidebb utak fáját! (Negatív élköltségek is megengedettek.) Mekkora a műveletigénye? Miért? Szemléltesse a működését az alábbi gráfon, ahol a csúcsok topologikus rendezése $\langle$a,b,c,d,e,f$\rangle$, és a b a startcsúcs! A legrövidebb utak fáját rajzolja is le! (20p)

a $\to$ d, 2; f, 3. b $\to$ c, 2; e, 4. c $\to$ d, -1; e, 1. d $\to$ e, 2; f, 2. e $\to$ f, -2.

5. Definiálja a *KMP* algoritmus *next* függvényét, majd adja meg az $ABABADA$ mintán! Szemléltesse *KMP* algoritmussal e minta előfordulásainak keresését a $ABABADABABADABABABADABADABA$ szövegben! Mi a *next* függvény szerepe a keresés során? Mi a *KMP* algoritmus előnye, illetve hátránya – műveletigény szempontjából – a *Quick-search* mintaillesztő algoritmussal összehasonlítva? (20p)