Csima Judit
csima@cs.bme.hu

5. gyakorlat
Dijkstra algoritmus, kupac, rendezések, bináris keresés

1. Egy irányított gráf csúcshalmaza $\{a, b, c, d, e, f\}$, az élek és súlyaik pedig az alábbiak: $s(a, b) = 5$, $s(a, e) = 6$, $s(b, c) = 4$, $s(b, d) = 6$, $s(c, a) = 3$, $s(c, d) = 1$, $s(d, e) = 2$, $s(e, c) = 2$, $s(e, f) = 1$, $s(f, b) = 3$, $s(f, c) = 1$, $s(f, d) = 1$.
   a) Dijkstra módszerével határozza meg a-ból az összes többi csúcsba vezető legrövidebb út hosszát. (Indokolni nem kell, de látszódjon, lépésenként hogyan változik a távolságokat tároló D tömb és a KÉSZ halmaz.)
   b) Egy él súlyát 1-gyel csökkentjük. Mely élek esetében nem változnak meg ezzel az a-tól mért távolságok?

2. (a) Építsen kupacot az órán tanult lineáris idejű módszerrel az alábbi tömbből: $31, 6, 50, 7, 2, 51$.
   (b) Szúrja be az így kapott tömbbe az 1, majd ezután az 5 számot.
   (c) Hajtson végre két egymást követő MINTÖR-t az így kapott kupacon.

3. Rendezze összefésüléses rendezéssel az alábbi tömböt: $3, 12, 1, 34, 6, 4, 0$.

---

4. Adjuk meg az összes olyan minimális élszámú irányított gráfot (élsúlyokkal együtt), amely(ek)re az alábbi táblázat a Dijkstra–algoritmusban szereplő D[ ] tömb változásait mutathatja. Adja meg a legrövidebb utakat tartalmazó P[ ] tömb állapotait is.

v_1	v_2	v_3	v_4	v_5	v_6
0	2	6	∞	∞	7
0	2	5	9	∞	6
0	2	5	6	9	6
0	2	5	6	8	6
0	2	5	6	7	6

5. A $G = (V, E)$ irányított gráfban a csúcsok egy része fontos, ezeknek a csúcsoknak a halmaza az $\emptyset \neq F \subseteq V$. A gráf minden éléhez tartozik egy pozitív élsúly. Az $u \in F$ fontos csúcs távolsága a $v \in F$ fontos csúcstól a legrövidebb olyan u-ból v-be menő út hossza, aminek nincs u-tól és v-től különböző fontos csúcsa. Legyen a gráf a mátrixával adott, és minden csúcsra adott az is, hogy fontos csúcs-e. Adjon algoritmust ami $O(|V|^2|F|)$ lépésben meghatározza az összes fontos csúcspár közötti távolságot!

6. Rendezze az $7, 3, 12, 1, 5, 4$ tömböt (a) beszúrásos rendezéssel, (b) összefésüléses rendezéssel, és (c) kupacos rendezéssel

7. Adjunk hatékony algoritmust egy kupac tizedik legkisebb elemének a megtalálására. Elemezzük a módszer költségét.

8. Legyen adott egy egészekből álló $A[1 : n]$ tömb valamint egy b egész szám. Szeretnénk hatékonyan eldönteni, hogy van-e két olyan $i, j \in \{1, \ldots, n\}$ index, melyekre $A[i] + A[j] = b$. Oldjuk meg ezt a feladatot $O(n \log n)$ időben!

---

9. Vidéken autózunk, ahol benzinkút csak bizonyos falvakban van. Az A falubeli benzinkúttól indulunk és a B faluba akarunk elérni (ahol szintén van benzinkút). A falvak közötti utakat egy n csúcsú e élű, összefüggő, irányítatlan gráf írja le, melynek csúcsai a falvak, az élek pedig a falvak közötti utakat jelentik, egy él súlya a két falut összekötő útszakasz hossza. A gráf az éllistájával adott, és ezen kívül adott még az a k falu, amelyben van benzinkút. Adjon $O(ke \log n)$ lépésszámú algoritmust, amely meghatározza az A-ból B-be vivő legrövidebb olyan útvonalat, melynek során soha nem kell 600 kilométernél többet autóznunk két benzinkút között. (ZH 2006. ápr. 7.)

10. Egy irányított gráf csúcshalmaza $\{a, b, c, d, e, f\}$, az élek és súlyaik pedig a következőek: $s(a, b) = 6$, $s(a, c) = 5$, $s(a, e) = 8$, $s(b, a) = 5$, $s(b, e) = 1$, $s(b, f) = 2$, $s(c, b) = 2$, $s(c, f) = 4$, $s(e, b) = 6$, $s(e, d) = 3$, $s(f, d) = 1$, $s(f, e) = 1$.
    a) Dijkstra-algoritmussal határozza meg a-ból az összes többi pontba vezető legrövidebb út hosszát. (Indokolni nem kell, de lépésenként írja fel a távolságokat tartalmazó D tömb és a KÉSZ halmaz állapotát.)
    b) Vegyük hozzá a gráfhoz az (b, d) élet. Milyen $s(b, d) \geq 0$ súlyok esetén változnának meg ezzel a legrövidebb utak hosszai?
    c) Egy él súlyát 1-gyel csökkentjük (az eredeti gráfban, ahol (b, d) él még nincsen). Mely élek esetében nem változnak meg ezzel az a-tól mért távolságok?

11. (a) Építsen kupacot az órán tanult lineáris idejű módszerrel az alábbi tömbből: $4, 3, 5, 21, 2, 7, 12, 6$.
    (b) Szúrja be az így kapott tömbbe az 1 számot.
    (c) Hajtson végre két egymást követő MINTÖR-t az így kapott kupacon.