Legrövidebb utak keresése; Kupac

Algoritmuselmélet
4. gyakorlat

2013. március 8.

1. 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

2. Adj $O(n^4)$ futási idejű algoritmust, amely egy mátrix segítségével adott n pontú irányítatlan, nemnegatív élsúlyokkal ellátott gráfban megtalálja a legrövidebb összhosszúságú kört (ami egy ponton nem mehet át kétszer).

3. A mátrixával adott G irányított gráf élei között van egy negatív súlyú él, a többi él súlya pozitív. A gráfban nincs negatív súlyú kör. Adjon $O(n^2)$ lépésszámú algoritmust az $s \in V(G)$ pontból az összes többi pontba vezető legrövidebb utak meghatározására.

4. 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.

5. Éllistával adott az n pontú $G(V, E)$ gráf, melynek minden e éle egy $c(e) > 0$ élsúllyal van ellátva. Egy adott $s \in V$ csúcsból akarunk egy adott $t \in V$ csúcsba eljutni a legolcsóbb módon, de az út költségét a szokásostól eltérően számoljuk: ha az e él az út s-től számított k-adik éle, akkor $k \cdot c(e)$ költséggel járul hozzá az út költségéhez. Adjon algoritmust, ami az ilyen értelemben vett legolcsóbb út költségét $O((n(n + |E|) \log n)$ lépésben.

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

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

8. Igazoljuk, hogy egy n elemből álló bináris kupac felépítése $\Omega(n)$ összehasonlítást igényel!

9. A kezdetben üres kupacba egyenként szúrunk be n elemet. Igazolja, hogy előfordulhat, hogy a beszúrások során végzett összehasonlítások száma $\Omega(n \log n)$.

10. Egy rendezett halmazból n elem kupacban van elhelyezve. Bizonyítsuk be, hogy a legnagyobb elem megkereséséhez $\Omega(n)$ összehasonlítás szükséges!

11. Egy kupacba beraktunk egy új x elemet, majd végrehajtottunk egy MINTÖR műveletet. Mikor fordul elő, hogy végül az eredeti kupacot kapjuk vissza?