

Minimális feszítőfák

- Hogyan működik Prim, Kruskal illetve Borůvka módszere minimális feszítőfák keresésére? Mit tudunk az algoritmusok lépésszámáról egy n csúcsú, e élű gráfban? Mi a köze a piros-kék algoritmusnak a fenti módszerekhez? Mikor takaros egy színezés?
- Éllistával adott egy G gráf, melynek n csúcsa és e éle van. A gráf minden csúcsához hozzá van rendelve egy 1 és k közötti egész szám (címke). Találjunk (ha létezik) olyan *tarka* utat a gráfban, amelyben minden $1 \leq i \leq k$ címke pontosan egyszer fordul elő. Az algoritmus lépésszáma legyen $O(k!(e + n))$.
- Egy irányított G gráfban hagyjuk el a források és a nyelőket. A maradék gráfban ismét hagyjuk el a forrásokat és a nyelőket, ezt ismételjük, amíg lehet. Bizonyítsuk be, hogy akkor és csak akkor kapunk üres gráfot, ha G -ben nincs irányított kör!
- Mátrixával adott egy G irányítatlan súlyozott gráf. Adott még a G -nek egy F minimális súlyú feszítőfája, és az F -nek egy f éle. Adjunk $O(n^2)$ lépésszámú algoritmust, ami meghatározza, hogy az f él súlyát meddig lehet úgy felemelni, hogy az F a gráf minimális feszítőfája maradjon.
- G irányítatlan gráf a következő éllistával (zárójelben jelölve a súlyokat, az éleket mindkét irányból felsorolva):
 $a: b(2), c(3); \quad b: a(2), d(3); \quad c: a(3), d(2); \quad d: b(3), c(2), e(4), f(2);$
 $e: d(4), f(1), g(2); \quad f: d(2), e(1), g(1), h(3); \quad g: e(2), f(3), h(4); \quad h: f(3), g(4);$
 - Prim módszerével minimális költségű feszítőfát! (a -ból indulva)
 - Kruskal módszerével minimális költségű feszítőfát!
 - Borůvka módszerével minimális költségű feszítőfát! (a -ból, e -ből és h -ból indulva)
 - a -ból kiinduló mélységi feszítőfát!
 - a -ból kiinduló szélességi feszítőfát!
- A szoftverpiacon n féle grafikus formátum közötti oda-vissza konverzióra használatos programok kaphatók: az i -edik és a j -edik között oda-vissza fordító program ára a_{ij} , futási ideje pedig t_{ij} (ha létezik).
 - Javasoljunk módszert annak megtervezésére, hogy minden egyes formátumról a saját grafikus terminálunk által megértett formátumra a lehető leggyorsabban konvertáljunk! (Az ár nem számít.)
 - Javasoljunk módszert annak eldöntésére, hogy mely programokat vásároljuk meg, ha azt szeretnénk a lehető legolcsóbban megoldani, hogy a megvett programok segítségével bármelyik formátumról bármelyik más formátumra képesek legyünk konvertálni. (Itt a futási idő nem számít).
- Legyen adva egy (egyszerű, irányítatlan, összefüggő) n pontú G gráf éllistával, az élek súlyozásával együtt. Tegyük fel, hogy a G -ből a v_1 csúcs, valamint a v_1 -re illeszkedő élek elhagyásával keletkező G' gráf még mindig összefüggő, és adott G' egy minimális költségű feszítőfája. Adjunk minél hatékonyabb algoritmust a G gráf egy minimális költségű feszítőfájának az elkészítésére! (Teljes értékű megoldás: $O(n \log n)$ idejű algoritmus.)
- Éllistával adott egy összefüggő, egyszerű, irányítatlan G gráf csupa különböző élsúlyokkal. Jelöljük n -nel a csúcsok, e -vel pedig az élek számát. Mutassunk egy lineáris (azaz $O(e)$) uniform költségű algoritmust, ami a G gráf egy minimális feszítőfájának $\lceil 2n/3 \rceil$ élét előállítja! (Egy olyan $\lceil 2n/3 \rceil$ elemszámú élhalmazt keresünk, ami biztosan része egy minimális költségű feszítőfának.)
- Legyen adott egy $n \times n$ pixelből álló fekete-fehér kép. Szeretnénk a képen a bal felső saroktól a jobb alsó sarokig egy jobbra-lefele haladó határvonalat húzni úgy, hogy a vonaltól jobbra-felfele eső fekete, valamint a vonaltól balra-lefele eső fehér pixelek számának az összege a lehető legkisebb legyen. Oldjuk meg ezt a feladatot $O(n^2)$ időben!