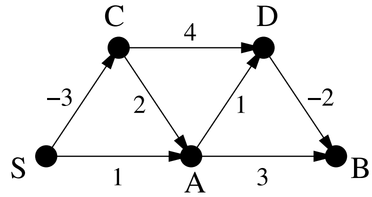
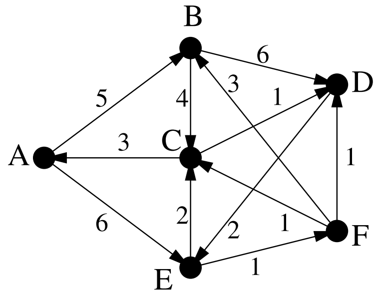
1. Határozzuk meg a Ford-algoritmus segítségével a jobbra látható gráfban az S pontból a többi pontba vezető legrövidebb utak hosszát és adjunk meg egy S-ből B-be vezető legrövidebb utat. Az algoritmus futtatásakor az éleket ábécé szerint növekvően sorszámozzuk: $A \to B$, $A \to D$, ..., $S \to C$.



2. a) Határozzuk meg a Dijkstra-algoritmus segítségével az A csúcsból a többibe vezető legrövidebb utak hosszát a jobbra látható gráfban és adjunk meg egy A-ból D-be vezető legrövidebb utat.

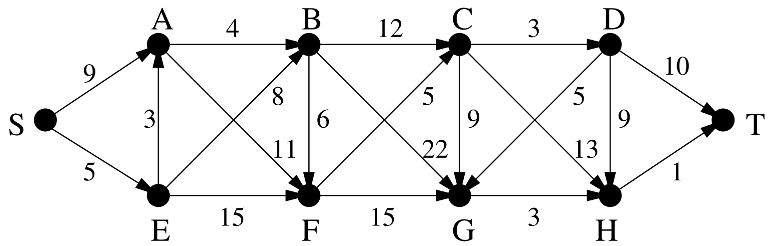
b) Valamely él súlyát 1-gyel csökkentjük. Mely élek esetében nem változnak meg ezzel az A-tól mért távolságok?

c) Vegyük hozzá a gráfhoz a $B \to E$ élt $t \geq 0$ élsúllyal. A t mely értékeire változnának meg ezzel a legrövidebb utak hosszai?



---

3. Határozzuk meg az S csúcsból a többibe vezető legrövidebb utak hosszát az alábbi gráfban és adjunk meg egy S-ből T-be vezető legrövidebb utat. A feladatot a tanult algoritmusok közül az erre a célra legalkalmasabbal oldjuk meg.



4. A **2**. feladat gráfjában változtassuk a C-ből A-ba menő él súlyát 3-ról (-3)-ra. Alkalmazhatjuk-e ekkor a Ford-algoritmust a B-ből a többi csúcsba vezető legrövidebb utak megtalálására? Ha igen, hajtsuk is végre az algoritmust és adjunk meg egy B-ből F-be vezető legrövidebb utat. Az algoritmus futtatásakor az éleket ábécé szerint növekvően sorszámozzuk: $A \to B$, $A \to E$,..., $E \to F$.

5. a) Nyári utazásunkra valutát akarunk váltani. A pénzváltó n különböző valutával foglalkozik, a j. fajta 1 egységéért r_{ij}-t kell fizetni az i. pénznemben. (Például ha a j. az Euró, az i. a Forint, akkor r_{ij} értéke most 314 körül lehet.) Adjunk olyan hatékony (vagyis polinomiális lépésszámú) algoritmust, ami az r_{ij} tömb felhasználásával meghatározza, hogy egy adott fajta valuta (például a Forint) birtokában az összes többit milyen legkedvezőbb átváltási aránnyal vásárolhatjuk meg, ha az átváltás történhet több lépcsőben is.

b) Mit mondhatunk az a) feladatban adott algoritmus lépésszámáról: milyen k esetén igaz, hogy az legföljebb $c \cdot n^k$ lépésben megáll (ahol c valamilyen konstans)?

c) Módosítsuk a feladatot annyiban, hogy nem csak egy adott valutáról kell meghatároznunk az összes többire való legkedvezőbb átváltási arányt, hanem bármely kettő között. Természetesen most is lehetséges volna az a) feladatban megadott algoritmust használni, ha azt n-szer lefuttatjuk (minden kiinduló valutára egyszer-egyszer). Adjunk ennél hatékonyabb algoritmust erre a feladatra.

---

6. a) A **2**. feladat gráfjában változtassuk az $E \to F$ él súlyát (-1)-re, az $F \to B$ él súlyát pedig (-4)-re. Alkalmazzuk a Ford-algoritmust az A-ból a többi csúcsba vezető legrövidebb utak hosszának meghatározására és adjunk meg egy A-ból C-be vezető legrövidebb utat. (Az éleket ismét ábécé szerint növekvően sorszámozzuk.)

b) Határozzuk meg a Dijkstra-algoritmus segítségével a **2**. feladat gráfjában a C csúcsból a többibe vezető legrövidebb utak hosszát és adjunk meg egy C-ből B-be vezető legrövidebb utat.

c) Vegyük hozzá az **1**. feladat gráfjához a $B \to S$ élt 2 súllyal. A kapott gráfban határozzuk meg minden lehetséges (u, v) csúcspárra az u-ból v-be vezető legrövidebb út hosszát a Floyd-algoritmus segítségével. Az algoritmus futtatásakor a csúcsokat ábécé szerinti növekvő sorrendben sorszámozzuk.

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

8. Adott egy $n \times n$ pixelből álló fekete-fehér kép. Szeretnénk a képen a bal felső saroktól a jobb alsó sarokig egy jobbra-lefelé haladó határvonalat húzni úgy, hogy a vonaltól jobbra-felfelé eső fekete, valamint a vonaltól balra-lefelé eső fehér pixelek számának összege a lehető legkisebb legyen. (A vonal mindenütt a pixelek között fut.) Adjunk a feladatra legfeljebb n^4-nel arányos lépésszámú algoritmust.