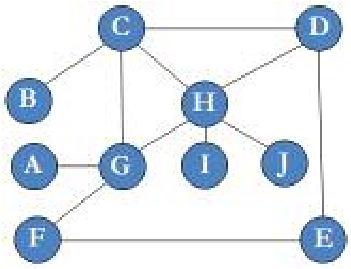
A

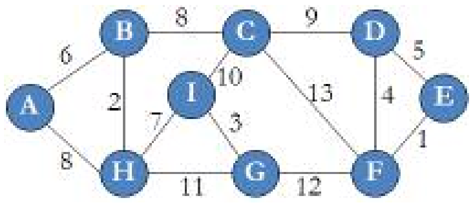
1a. feladat: 10

Adja meg a mellékelt gráf pontbejárási sorrendjét, ha a bejárás ***szélességi****! A bejárást kezdje a '****C****'-vel! Az egyes pontokból az óra járásával ellentétes irányú az élsorrend („éjféllel" befejezve).*
Pontozás: minden helyes gráfpontért 1 pont, az első hibáig, utána 0 pont!



2a. feladat: 14

Adja meg a mellékelt gráf minimális feszítőfáját a ***Kruskal*** *módszer szerint! Egy-egy ábrával szemléltesse az algoritmus működését: az elhangzott algoritmus második ciklusának minden egyes lefutásához tartozzon egy-egy –állapotot kifejező– ábra.*
Pontozás: minden helyes ábráért 1 pont, az első hibáig, utána 0 pont!



3a. feladat: 15

Egy gráf ***párosság****ának megállapítására a szélességi bejárás algoritmusából indultunk ki az előadáson. Írja meg ugyanezen feladat megoldására azt a megoldásváltozatot, amely alapjául a* ***mélységi bejárás*** *szolgál!*

4a. feladat: 15

Adott egy térkép, amelyen városok és a városokat összekötő utak hálózata látható. Az utak (a városok között) kétirányúak. A városokat egy-egy pozitív egész számmal jellemezzük. Adjon algoritmust a következő feladat megoldására! Adott a kezdő- és a célváros. Határozzunk meg egy olyan utat a kezdő- és a végváros között, amelyre a várost jellemző számok összege a lehető ***legkisebb****! (Az egyes városokat legfeljebb egyszer érinthetjük!)*

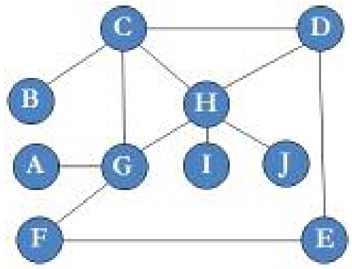
Tervezett ponthatárok:

Kettes, ha legalább:	*28*	*Négyes, ha legalább:*	*41*
Hármas, ha legalább:	*34*	*Ötös, ha legalább:*	*48*
	Maximum:	*10+14+15+15=54*	

B

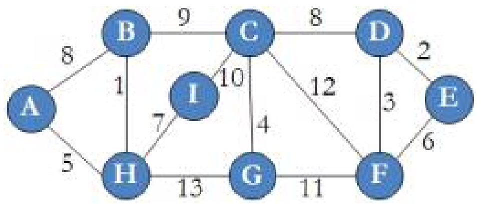
1b. feladat: **10**

*Adja meg a mellékelt gráf pontbejárási sorrendjét, ha a bejárás **mélységi**! A bejárást kezdje a 'H'-val! Az egyes pontokból az óra járásával ellentétes irányú az élsorrend („éjféllel" befejezve). A visszalépés során érintett pontokat zárójelek közé zárva jelezze!*
Pontozás: minden helyes gráfpontért 1 pont, az első hibáig, utána 0 pont!



2b. feladat: **14**

*Adja meg a mellékelt gráf minimális feszítőfáját a **Kruskal** módszer szerint! Egy-egy ábrával szemléltesse az algoritmus működését: az elhangzott algoritmus második ciklusának minden egyes lefutásához tartozzon egy-egy –állapotot kifejező– ábra.*
Pontozás: minden helyes ábráért 1 pont, az első hibáig, utána 0 pont!



3b. feladat: **15**

*Egy gráf **párosság**ának megállapítására a szélességi bejárás algoritmusából indultunk ki az előadáson. Írja meg ugyanezen feladat megoldására azt a megoldásváltozatot, amely alapjául a **mélységi bejárás** szolgál!*

4b. feladat: **15**

*Adott egy térkép, amelyen városok és a városokat összekötő utak hálózata látható. Az utak (a városok között) kétirányúak. A városokat egy-egy pozitív egész számmal jellemezzük. Adjon algoritmust a következő feladat megoldására! Adott a kezdő- és a célváros. Határozzunk meg egy olyan utat a kezdő- és a végváros között, amelyre a várost jellemző számok összege a lehető **legkisebb**! (Az egyes városokat legfeljebb egyszer érinthetjük!)*

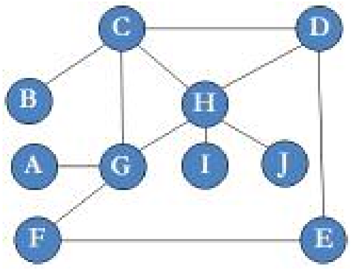
Tervezett ponthatárok:

Kettes, ha legalább:	*28*	*Négyes, ha legalább:*	*41*
Hármas, ha legalább:	*34*	*Ötös, ha legalább:*	*48*
	Maximum:	*10+14+15+15=54*	

Megoldas

1a. feladat: **10**

*Adja meg a mellékelt gráf pontbejárási sorrendjét, ha a bejárás **szélességi**! A bejárást kezdje a '**C**'-vel! Az egyes pontokból az óra járásával ellentétes irányú az élsorrend („éjféllel" befejezve).*
Pontozás: minden helyes gráfpontért 1 pont, az első hibáig, utána 0 pont!

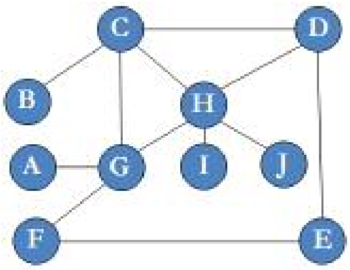


Megoldás:

CBGHDAFIJE

1b. feladat: **10**

*Adja meg a mellékelt gráf pontbejárási sorrendjét, ha a bejárás **mélységi**! A bejárást kezdje a '**H**'-val! Az egyes pontokból az óra járásával ellentétes irányú az élsorrend („éjféllel" befejezve). A visszalépés során érintett pontokat zárójelek közé zárva jelezze!*
Pontozás: minden helyes gráfpontért 1 pont, az első hibáig, utána 0 pont!

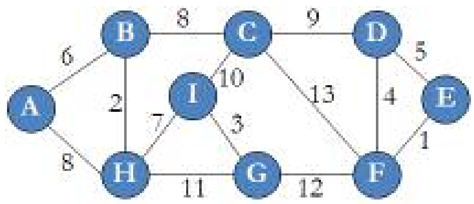


Megoldás:

HCB(C)GA(G)FED(EFGCH)I(H)J

2a. feladat: **14**

*Adja meg a mellékelt gráf minimális fesztőfáját a **Kruskal** módszer szerint! Egy-egy ábrával szemléltesse az algoritmus működését: az elhangzott algoritmus második ciklusának minden egyes lefutásához tartozzon egy-egy –állapotot kifejező– ábra.*
Pontozás: minden helyes ábráért 1 pont, az első hibáig, utána 0 pont!

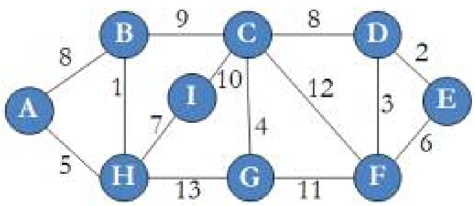


Megoldás:

Az egyes lépésekben érintett éleket sorolom föl; zárójelek között a sorra kerülő, de nem kiválasztott él:
1 EF; 2 BH; 3 GI; 4 DF; (5 DE); 6 AB; 7 HI; (8 AH); 8 BC; 9 CD;
(10 CI); (11 HG); (12 FG); (13 CF)

2b. feladat: **14**

*Adja meg a mellékelt gráf minimális feszítőfáját a **Kruskal** módszer szerint! Egy-egy ábrával szemléltesse az algoritmus működését: az elhangzott algoritmus második ciklusának minden egyes lefutásához tartozzon egy-egy –állapotot kifejező– ábra.*
Pontozás: minden helyes ábráért 1 pont, az első hibáig, utána 0 pont!



Megoldás:

Az egyes lépésekben érintett éleket sorolom föl; zárójelek között a sorra kerülő, de nem kiválasztott él:
1 HB; 2 DE; 3 DF; 4 CG; 5 AH; (6 EF); 7 HI; (8 AB); 8 CD; 9 BC;
(10 CI); (11 FG); (12 CF); (13 HG)

3a/b. feladat: **15**

*Egy gráf **párosság**ának megállapítására a szélességi bejárás algoritmusából indultunk ki az előadáson. Írja meg ugyanezen feladat megoldására azt a megoldásváltozatot, amely alapjául a **mélységi bejárás** szolgál!*

Megoldás:

```
[párosságvizsgálat:
 eltérés a Mikrológiás változattól:
 a pontHalmaz-ra nincs szükség, mivel a belőle szedett
 információ kinyerhető az oszt vektorból is;
 a szélességi bejárás helyett a mélységire építjük.]
Változó oszt:Tömb(1..maxP:Egész)  [a gráf pontjainak osztály besorolása]
        parosE:Logikai
        eH:TElHalmaz  [élek halmaza]
        v:TPontVerem  [pontok verme]
        p,sp:TPont  [segédpont]
        i,pi:Egész
Ciklus i=1-től maxP-ig
  oszt(i):=0  [egyik pont sincs még besorolva]
Ciklus vége
pi:=1;  p:=pontok(pi)
TPontVeremUres(v);  TPontVerembe(v,p)
oszt(p):=1 [1 pont];  parosE=Igaz [1 pont]
Ciklus amíg nem TPontVeremUresE(v) és parosE [1 pont]
  TPontVeremTeto(v,p)
  ElsoSzomszed(eH,p,sp)
  pi:=1
  Ciklus amíg pi<SzomszedPontokSzama(eH,p) és oszt(sp)=3-oszt(p)
    KovetkezoSzomszed(eH,p,sp,sp)
    pi:+1
  Ciklus vége
  Ha oszt(sp)=0 akkor  [nem jártunk még sp-nél]
    TPontVerembe(v,sp);  oszt(sp):=3-oszt(p) [6 pont]
  különben
    TPontVerembol(v,p)
  Elágazás vége
  parosE:=oszt(sp)≠oszt(p) [6 pont]
Ciklus vége
```

Megjegyzés [SzP1]: Halmazos megoldásnál itt más (ÜresHalmaz) szerepel, a mélységi bejárás alapváltozatának megfelelően

Megjegyzés [SzP2]: Halmazos megoldásnál itt más (ElemeE) szerepel, a mélységi bejárás alapváltozatának megfelelően

Megjegyzés [SzP3]: Halmazos megoldásnál itt más (nem ElemeE) szerepel, a mélységi bejárás alapváltozatának megfelelően

4a/b. feladat: **15**

*Adott egy térkép, amelyen városok és a városokat összekötő utak hálózata látható. Az utak (a városok között) kétirányúak. A városokat egy-egy pozitív egész számmal jellemezzük. Adjon algoritmust a következő feladat megoldására! Adott a kezdő- és a célváros. Határozzunk meg egy olyan utat a kezdő- és a végváros között, amelyre a várost jellemző számok összege a lehető **legkisebb**! (Az egyes városokat legfeljebb egyszer érinthetjük!)*

Megoldás:

Csapda: nem lehet (gondolkodás nélkül) a végpontbeli várost az él hosszának tekinteni, s így súlyozott gráfnak tekinteni. Mert nem egyértelmű a hossz-hozzárendelés a két végpontba elhelyezett, esetleg különböző érték miatt.

1. ötlet:

*A **szélességi bejárás**ra építhető. A következők figyelembe vételével:*

1. *Az egyes pontokba jutva fel kell jegyezni a „**honnan**”-t és az „**odáig**” számolt összértéket.*
2. *Ha **már meglátogatott pont**hoz érünk, akkor meg kell vizsgálni, hogy ez a (több lépéses) út összhossza **jobb-e**. Ha igen, akkor ezt kell választani, azaz a „**honnan**” és az „**odáig**”értéket **módosítani** kell.*
3. *Mivel többlépéses út is lehet kisebb összértékű, azért a **végpontba jutás nem** terminálja a ciklust.*

2. ötlet:

*Súlyozott gráf helyett azonban a súlyozott gráfra alkalmazott legrövidebb út algoritmus mégis alkalmazható, ha az ÉlHossz fv gyanánt az aktuális **él végpontjában tárolt érték**et (Érték fv-t) használhatjuk.*

3. ötlet:

*Ha az élek hosszát végpontjaiban tárolt értékek **összeg**eként definiáljuk, akkor egy tetszőleges* K—V = K–A_1–A_2–…–A_u–V *út pont-összege:*

$$\textit{ÚtÖsszeg}(K–A_1–A_2–\ldots–A_u–V) = \text{Érték}(K) + \sum_{i=1}^{u} \text{Érték}(A_i) + \text{Érték}(V)$$

összefüggés helyett az

$$\textit{ÚtHossz}(K–A_1–A_2–\ldots–A_u–V) = \text{ÉlHossz}(K - A_1) + 2 * \sum_{i=1}^{u-1} \text{ÉlHossz}(A_i - A_{i+1}) + \text{ÉlHossz}(A_u - V)$$

lesz, ami sajnos eltér az elvárttól.

*De mivel nem az érték érdekel minket, hanem a **minimum értékű ut**at keressük, ezért valójában az utóbbi értelmű úthosszal definiált **súlyozott gráfra vonatkozó legrövidebb út keresés algoritmusa** is használható. A formulában a kezdő- és végpont egyszeres szorzó volta nem ront semmit, hiszen minden számunkra érdekes út ugyanott kezdődik és végződik, tehát **csak a közbülső pontok minimuma** számit (amely 2-szeres szorzója minden útra egyformán megvan).*