



A 2014/2015 tanévi Országos Középiskolai Tanulmányi Verseny második forduló javítási-értékelési útmutató

INFORMATIKA II. (programozás) kategória

Kérjük a tisztelt kollégákat, hogy az egységes értékelés érdekében az alábbi eljárást alkalmazzák:

1. Az értékelő gépen hozzák létre a \NT2 könyvtárat.
2. Bontsa ki az NT2.zip állományt.
3. Minden versenyző számára hozzanak létre egy külön könyvtárat, és ezekbe másolják be a versenyzők programjait (a feladatleírásban szereplő néven). (Ha nem adott be exe-t, akkor le kell fordítani!)
4. Egy versenyző értékelése:
 - A. Az aktuális könyvtár legyen a versenyző könyvtára.
 - B. Adják ki az \NT2\T3 (\nt2\t2) parancsot, amely lefuttatja a versenyző programjait minden tesztetresre. Ha a végrehajtás megszakad, vagy meg kell szakítani, mert letelt a 60 másodperc, akkor ismét a \NT2\T3 parancsot kell kiadni, mindaddig, amíg az "ÉRTÉKELES BEFEJEZŐDÖTT" üzenet meg nem jelenik a képernyőn. (A futtató tudja, hogy honnan kell folytatnia.). Ezt követően automatikusan elindul a megoldásokat értékelő program, amely összesítést készít a versenyző könyvtárában EREDMENY.TXT néven, és az eredményt a képernyőre is kiírja.

1. feladat: Fenyőfa (30 pont)

Egy erdő négyzetrácsos területén N sorban, M oszlopban fenyőfák nőnek. Tudjuk minden rácsponttól, hogy ott van-e fenyőfa. Az erdő tulajdonosa pontosan K fát szeretne kivágni, de úgy, hogy egy téglalap alakú területen az összes fát kivágja.

Készíts programot (`fenyo.pas`, ...), amely megadja egy olyan legkisebb téglalap alakú terület bal felső és jobb alsó sarkát, amelyen pontosan K darab fenyőfa van!

A `fenyo.be` szöveges állomány első sorában a terület sorainak és oszlopainak száma ($1 \leq N, M \leq 100$), valamint a kivágandó fenyők száma ($1 \leq K \leq 1000$) van, egy-egy szóközzel elválasztva. A következő N sor mindegyikében M szám található: az i -edik sor j -edik eleme 1, ha van fenyőfa az (i, j) pozíción, 0 egyébként.

A `fenyo.ki` szöveges állomány egyetlen sorába egy legkisebb olyan terület bal felső és jobb alsó sarkának sor- és oszlopindexét kell írni, amelyben pontosan K darab fenyőfa van! Ha nincs ilyen terület, akkor négy darab -1-est kell kiírni!

Példa:

<code>fenyo.be</code>	<code>fenyo.ki</code>
4 6 3	3 2 3 5
1 1 0 0 0 0	
0 0 0 0 0 0	
0 1 1 0 1 1	
1 0 0 1 0 0	

Értékelés:

- | | |
|----------------------------------------------------------------|--------|
| 1. Nincs ilyen terület | 1 pont |
| 2. Egyetlen, pontosan K méretű terület, 1 sorban | 3 pont |
| 3. Egyetlen pontosan K méretű terület, 1 oszlopban | 3 pont |
| 4. Egyetlen pontosan K méretű terület, téglalapban | 3 pont |
| 5. 1 soros terület, nem az elsőnek kiválasztott a legkisebb | 3 pont |
| 6. 1 oszlopos terület, nem az elsőnek kiválasztott a legkisebb | 3 pont |

7. Téglalap alakú terület, nem az elsőnek kiválasztott a legkisebb	3 pont
8. Véletlen közepes teszt	3 pont
9. Véletlen nagy teszt, sűrűn fákkal	4 pont
10. Véletlen nagy teszt, ritkán fákkal	4 pont

2. feladat: Következő feladatbeosztás (30 pont)

Egy iskolában M különböző feladatot osztanak ki, egy tanuló azonban legfeljebb csak egy feladatot kaphat. Az iskola N tanulójaiból kell kiválasztani a feladatokat elvégzőket. A lehetséges feladat-kiosztások szám M -esek, amelyek mindegyik tagja 1 és N közötti egész szám. A lehetséges szám M -esek sorba rendezhetők lexikografikusan, pl. $N=3, M=2$ esetén:

1 2, 1 3, 2 1, 2 3, 3 1, 3 2.

Készíts programot (`feladat.pas, ...`), amely egy adott feladat kiosztásra megadja a ciklikusan lexikografikusan következőt! (Megjegyzés: a fenti példában a 3 2 után az 1 2 a következő.)

A `feladat.be` állomány első sorában a tanulók száma ($1 \leq N \leq 100$) és a feladatok száma ($1 \leq M \leq 10$), van egy szóközzel elválasztva. A következő sorban pontosan M különböző szám van, egy feladat kiosztás ($1 \leq S_i \leq N$).

A `feladat.ki` állomány egyetlen sorába a bemenetbelire következő feladat kiosztást kell írni!

Példa:

```
feladat.be                feladat.ki
4 3                      1 3 2
1 2 4
```

Értékelés:

1. $N=M$, elsőre következő permutáció	2 pont
2. $N=M$, utolsóra következő permutáció	2 pont
3. $N=M$, általános eset	2 pont
4. 1-2-x típusú esett, x növekszik	3 pont
5. 1-x-N típusú eset	3 pont
6. x-...-x-2 típusú eset, nincs benne 1 és 3 és N	3 pont
7. x-...x-y típusú eset, minden y -nál nagyobb szerepel az x -ek között	3 pont
8. x-...-N-(N-1) eset	3 pont
9. x-...-(N-1)-N eset, 1 nem szerepel az x -ek között	3 pont
10. Véletlen közepes teszt	3 pont
11. Véletlen nagy teszt	3 pont

3. feladat: Vasút (30 pont)

Bergengócia vasúthálózata olyan, hogy bármely városból bármely másik városba csak egyféleképpen lehet eljutni. Minden vonat a fővárosból (1-es sorszámú város) indul és valamely olyan városig megy, ahonnan már nincs tovább vasúti pálya. Két város között a vonatút hossza a köztük levő vasútállomások száma+1!

Készíts programot (`vasut.pas, ...`), amely megadja a leghosszabb vonatút hosszát, ahol a vasút nem ágazik el!

A `vasut.be` állomány első sorában a városok száma van ($2 \leq N \leq 10\ 000$), a következő $N-1$ sorban pedig az egyes vasútszakaszok leírása ($1 \leq A_i, B_i \leq N$), ami azt jelenti, hogy van A_i -ből B_i -be menő közvetlen vasúti összeköttetés. A vasúti pálya bármely városból legfeljebb 10-felé ágazhat.

A `vasut.ki` állomány első sorába a leghosszabb vonatot hosszát kell írni, ahol a vasút nem ágazik el!

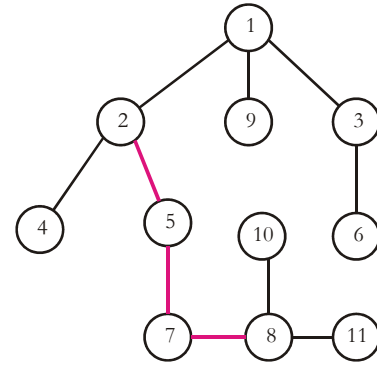
Példa:

`vasut.be`

```
11
1 2
1 3
2 4
2 5
1 9
3 6
5 7
7 8
8 10
8 11
```

`vasut.ki`

3



Értékelés:

- | | |
|------------------------------------|--------|
| 1. Egyetlen lánc | 2 pont |
| 2. Van lánc 1-ből | 3 pont |
| 3. Mindenhol van elágazás | 3 pont |
| 4. Lánc az i. elemtől levélig | 3 pont |
| 5. Lánc 1.ből elágazásig | 3 pont |
| 6. Lánc az i. elemtől elágazásig | 3 pont |
| 7. Lánc nem a leghosszabb úton | 3 pont |
| 8. Véletlen közepes teszt (N=100) | 3 pont |
| 9. Véletlen nagy teszt (N=1000) | 3 pont |
| 10. Véletlen nagy teszt (N=10 000) | 4 pont |

4. feladat: Szállítás (30 pont)

Egy raktárból K kamionnal kell elszállítani tárgyakat. Minden kamion azonos S kapacitású, ami azt jelenti, hogy legfeljebb S összsúlyú tárgy rakható rá. A tárgyak a raktárban egyetlen sorban helyezkednek el, ezért az aktuális kamionra csak a sor két végéről lehet felrakni tárgyat.

Készíts programot (`szallit.pas,...`), amely kiszámítja, hogy legjobb esetben hány tárgyat lehet elszállítani a K kamionnal!

A `szallit.be` szöveges állomány első sora a tárgyak számát ($1 \leq N \leq 10\,000$), a kamionok számát ($1 \leq K \leq 100$) és a kamionok közös kapacitását ($1 \leq S \leq 1000$) tartalmazza. A második sor pontosan N pozitív egész számot tartalmaz egy-egy szóközzel elválasztva, az elszállítandó tárgyak súlyait ($1 \leq T_i \leq S$).

A `szallit.ki` szöveges állomány első és egyetlen sorába egy egész számot kell írni, a legtöbb elszállítható tárgy számát!

Példa:

`szallit.be`

```
8 3 100
20 70 10 30 80 60 50 30
```

`szallit.ki`

7

Értékelés:

1. A megoldás 1	1 pont
2. A megoldás 2	2 pont
3. Tárgyak súlya kicsi, $N=100, K=10$	3 pont
4. Tárgyak súlya kicsi, $N=200, K=100$	3 pont
5. Tárgyak súlya kicsi, $N=500, K=10$	3 pont
6. Tárgyak súlya kicsi, $N=1000, K=100$	3 pont
7. Véletlen kis bemenet	3 pont
8. Véletlen közepes bemenet	4 pont
9. Véletlen nagy bemenet	4 pont
10. Véletlen nagy bemenet	4 pont

5. feladat: Hálózat (30 pont)

Egy kommunikációs hálózat csomópontokból és csomópont párokat összekötő egyirányú átvitelt megvalósító közvetlen vonalakból épül fel. Azt mondjuk, hogy egy U csomópontból elérhető a V csomópont, ha U -ból lehet átvitelt megvalósítani V -be esetleg közbülső csomópontokon keresztül. A hálózatnak van egy K kitüntetett csomópontja. Az üzemeltetés szeretné tudni, hogy melyek azok a K -tól különböző X csomópontok, amelyek legfeljebb T közbülső csomópont érintésével elérhetők K -ból és X -ből akárhány csomópont érintésével elérhető K .

Írj programot (`halozat.pas,...`), amely kiszámítja azokat a csomópontokat, amelyek legfeljebb T közbülső csomóponton keresztül elérhetők K -ból, és amelyekből elérhető K !

A `halozat.be` szöveges állomány első sorában négy egész szám van egy-egy szóközzel elválasztva: a csomópontok száma ($2 \leq N \leq 10\ 000$), a közvetlen vonalak száma ($2 \leq M \leq 300\ 000$), a kitüntetett csomópont sorszáma ($1 \leq K \leq N$) és a kérdésben szereplő távolság ($0 \leq T < N$). A további M sor mindegyike két $U \ V$ egész számot tartalmaz egy szóközzel elválasztva, ami azt jelenti, hogy van az U csomópontból a V csomópontba átvitelt megvalósító közvetlen vonal ($1 \leq U \neq V \leq N$).

A `halozat.ki` szöveges állomány első sora egy egész számot tartalmazzon, azon csomópontok számát, amelyek legfeljebb T közbülső csomóponton keresztül elérhetők K -ból, és amelyekből elérhető K ! A második sor ezen csomópontok sorszámaát tartalmazza egy-egy szóközzel elválasztva, növekvő sorrendben.

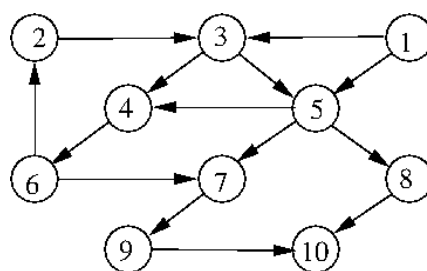
Példa:

`halozat.be`

```
10 14 3 2
2 3
1 3
3 4
3 5
1 5
4 6
6 2
6 7
7 9
9 10
5 8
8 10
5 4
5 7
```

`halozat.ki`

```
4
2 4 5 6
```



Értékelés:

1. Nincs ilyen pont	1+0 pont
2. $T=1, N=10$	1+1 pont
3. $T=2, N=20$	1+2 pont
4. $T=N-1, N=1000$	1+2 pont
5. Véletlen kis bemenet	1+2 pont
6. Véletlen kis bemenet	1+2 pont
7. Véletlen közepes bemenet	1+2 pont
8. Véletlen közepes bemenet	1+3 pont
9. Véletlen nagy bemenet	1+3 pont
10. Véletlen nagy bemenet	1+3 pont

Elérhető összpontszám: 150 pont + 50 pont az 1. fordulóból