

1. feladat: Vendégség (35 pont)

Egy rendezvényre sok vendéget hívtak meg. A szervezők feljegyezték minden vendég érkezési és távozási idejét. A szervezők szeretnék tudni, hogy ki az a vendég, aki a legkevesebb más vendéggel találkozott. Két vendég, akiknek érkezési idejük $a1$ illetve $a2$, távozási idejük pedig $b1$ illetve $b2$, akkor és csak akkor találkoztak, ha $a1 \leq b2$ és $a2 \leq b1$.

Készíts programot (**vendeg.pas, vendeg.c, ...**), amely kiszámítja, hogy ki az a vendég, aki a legkevesebb másik vendéggel találkozott!

A **vendeg.be** szöveges állomány első sorában egy egész szám van, a vendégek N száma ($1 \leq N \leq 1\,000\,000$). A további N sor mindegyike két egész számot tartalmaz egy szóközzel elválasztva, az első szám E a vendég érkezési, a második szám T pedig a vendég távozási ideje. ($1 \leq E < T \leq 100\,000$). A vendégeket az $1, \dots, N$ számokkal azonosítjuk. A bemenet $i+1$ -edik sorában van az i . vendég adata.

A **vendeg.ki** szöveges állomány első és egyetlen sora két egész számot tartalmazzon, egy szóközzel elválasztva, az első annak a vendégnek a sorszáma legyen, aki a legkevesebb másik vendéggel találkozott. A második szám pedig azon vendégek száma, akivel találkozott! Több megoldás esetén bármelyik megadható. Saját maga nem számít be a találkozott vendégek közé. Több megoldás esetén a legkisebb sorszámút kell megadni.

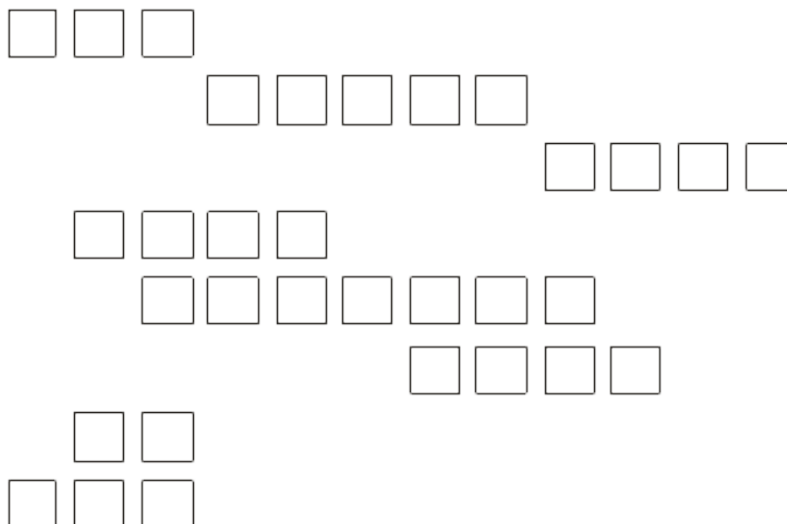
Példa:

vendeg.be

8
1 3
4 8
9 12
2 5
3 9
7 10
2 3
1 3

vendeg.ki

3 2



2. feladat: Kiegyensúlyozott keresőfa (30 pont)

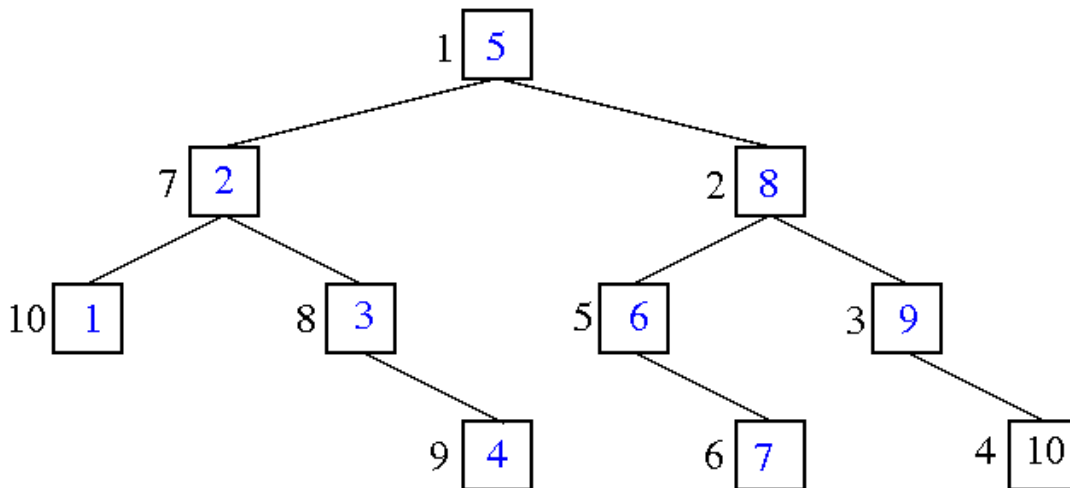
Rendezett adatsorozatban egy adatelemet bináris kereséssel is megkereshetünk. Ha gyakran kell ilyen keresést végezni, akkor elhelyezhetjük az adatokat bináris fa adatszerkezetben. Egy bináris fát keresőfának nevezünk, ha minden p pontjára teljesül, hogy ha a q pont p bal-részfájában van, akkor q -ban kisebb adat van, mint p -ben, és ha az r pont p jobb-részfájában van, akkor r -ben nagyobb adat van, mint p -ben. Bináris keresőfa esetén egy adat megkereséséhez legrosszabb esetben annyi összehasonlítást kell tenni, ahány pontja van a fának a leghosszabb, gyökértől levélig tartó úton. Ezért, érdemes kiegyensúlyozottá tenni a fát. Azt mondjuk, hogy egy bináris fa kiegyensúlyozott, ha bármely p pontjára teljesül, hogy a bal-részfájában és a jobb-részfájában levő pontok száma legfeljebb eggyel tér el.

Ha az $a_1 < a_2 < \dots < a_N$ rendezett sorozat elemeit kell elhelyezni egy bináris keresőfában, akkor elegendő megadni azt, hogy a fa minden pontjában melyik indexű elem van.

Készíts programot (**kerfa.pas**, **kerfa.c**, ...), amely elkészít egy N -pontú kiegyensúlyozott bináris keresőfát!

A **kerfa.be** szöveges állomány első és egyetlen sora egy egész számot tartalmaz, a bináris keresőfa pontjainak N ($1 \leq N \leq 20000$) számát. A keresőfa pontjait az $1, \dots, N$ számokkal azonosítjuk.

A **kerfa.ki** szöveges állomány pontosan N sort tartalmazzon, soronként három egész számot, szóközzel elválasztva. Az i -edik sorban a bináris keresőfa i . azonosítójú pontjának adatai legyenek. Az első szám a pontba helyezendő adatelem sorszáma legyen. A második szám a bal fiú sorszáma, a harmadik pedig a jobb fiú sorszáma. Ha a pontnak valamelyik fia hiányzik, akkor a 0 számot kell megadni. Több megoldás esetén bármelyik megadható.



Példa:

kerfa.be

10

kerfa.ki

```

5 7 2
8 5 3
9 0 4
10 0 0
6 0 6
7 0 0
2 10 8
3 0 9
4 0 0
1 0 0
  
```

3. feladat: Két gép (35 pont)

Egy vállalkozó alkatrészeket gyárt két gépén. Mindkét gép képes elvégezni ugyanazt a műveletet, azonban nem feltétlenül azonos idő alatt. N alkatrész vár megmunkálásra, mindegyikről tudjuk, hogy mennyi idő alatt lehet az elvégezni az A , illetve a B gépen. A vállalkozó szeretné tudni, hogy legjobb esetben mennyi idő alatt lehet elvégezni az összes alkatrészen a kívánt műveletet. Az egyes alkatrészeket tetszőleges sorrendben lehet feldolgozni.

Készíts programot (**ketgep.pas**, **ketgep.c**, ...), amely kiszámítja, hogy legjobb esetben mennyi idő alatt lehet elvégezni az összes alkatrészen a kívánt műveletet!

A **ketgep.be** szöveges állomány első sorában egy egész szám, az alkatrészek N száma ($1 \leq N \leq 5000$). A következő N sor mindegyikében két pozitív egész szám van, szóközzel elválasztva. Az első szám az az idő, amely alatt az A gép, a második pedig az az idő, amely alatt a B gép elvégzi az alkatrészen a szükséges műveletet. Minden idő értéke legfeljebb 100.

A **ketgep.ki** szöveges állomány első és egyetlen sorába egy egész számot kell írni, azt a legkisebb idő értéket, amely alatt el lehet végezni az összes alkatrészen a kívánt műveletet!

Példa:

ketgep.be

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

ketgep.ki

12