### Feladat

Egy fájlban négymilliárd 32 bites (int) egész számot tárolunk (gyakorlatilag mindegy, hogy pontosan milyen formában, szövegesen vagy binárisan), mindegyik szám legfeljebb egyszer szerepel, rendezetlenül. Adjunk algoritmust, amely megkeres egy tetszőleges olyan (nyilván 32 biten tárolható) számot, amely a fájlban nincs benne.

Többféle megoldás is lehetséges, így többféle szempont szerint is adj optimális algoritmust, pl kevés vagy sok memória áll rendelkezésre, gyors vagy lassú a diszk. A legjobb nyilván, ha korlátozott memóriát igénylő és kevés diszkművelettel dolgozó algoritmust is adsz. Mindegyik változat esetében add meg, hogy mennyi memóriát és hány diszkműveletet igényel a futás.

Megoldásként elég az algoritmus precíz szöveges megfogalmazása, nem szükséges kód vagy pszeudokód.

### Megoldás

Kiinduló helyzetként adott egy 4 milliárd számot tartalmazó fájl, melyek egyenként 32 biten tárolhatók. Ez 16GB-ot jelent (4×10^9×4 bytes). Feltételezve, hogy a memória mérete (memSizeInGb) a szűk keresztmetszet, szükséges szétdarabolni a fáljt 16 / memSizeInGb méretű *k* darab kisebb részre:

1. Kiveszünk egy memóriaméretü darabot a fájlból, rendezzük valamilyen alap rendezési módszerrel.
2. Ezután rendzett tömböt kiírjuk a diszkre.
3. Adding ismételjük az 1-es 2-es lépést, amíg az egész nagy fájl tartalma megtalálható a kis, rendezett fájlokban
4. Miután k darab rendezett számsorokból álló fájl áll rendelkezésre hajtsunk végre k-way merge-t, annyi módosítással, hogy nem szükséges kiirni a már rendezett tömböt, ezért lehetnek nagyobbak is a beolvasandó csonkok:
   1. Olvassunk be mindegyik fájlból egy memSizeInGb / (k) méretű részt. (Az egyikből hagyjunk ki az utolsót mert kell a hely a 2-es ponthoz)
   2. Vegyünk fel egy változót Integer.MIN\_VALUE értékkel
   3. Ekkor adott k darab tömb. Hasonlítsuk össze az első indexen lévő elemeket és vegyük ki a legkisebbet.
   4. Vizsgáljuk meg hogy a 2-es pontban felvett változónál mennyivel nagyobb. Ha 2 v annál több máris találtunk egy köztes elemet és egy lehetséges megoldását a feladatnak.
   5. Ha nincs köztes szám akkor a 2es pontban felvett változóba mentsük el a talált legkisebb elemet.
   6. Ismételjük addig a 3-as, 4-es 5-ös pontot amig nem találunk egy megoldást.
   7. Ha valamelyik a k darab tömbünk közül bármelyik is kiürül töltsük fel újra a hozzá tartozó fájlból.

Komplexitás:

1. diszkolvasás az 1es lépésben k db
2. diszkre kiírás a 2es lépésben k db
3. k-way merge alatt csak olvasni kell a diszkről. Ennek maximális száma: k^2  
   Azzal is érdemes számolni, hogy ha valamelyik, már rendezett darab összes eleme feldolgozásra került a 4es pontban, akkor a beolvasandó csonkok mérete is növekedhet, hiszen már memSizeInGb / (k-1) a megengedhető méret. Ez csökkenteni fogja a szükséges diszkolvasási mennyiséget.

Ez tekinthető kiindulópontban. Innetől lehet variálni és eltérni a k szám helyes meghatározásán, figyelembe véve a CPU sebességét, magok számát, a diszkre írás/olvasás sebességét, stb.