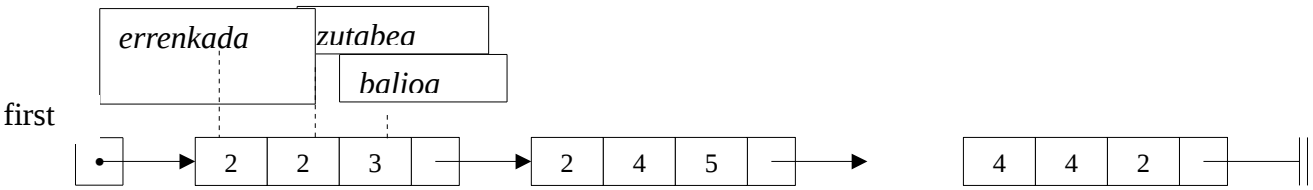
1.- Matrize batuketa (1,5 puntu)

Matrize sakabanatuetan posizio gehienetan zero balioa egoten da. Adibidez:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

Mota honetako matrizea zerrenda estekatu sinplea erabiliz adieraz daiteke. Matrizean 0 balioa ez duten elementuak baino ez dira sartzen zerrenda estekatuan. Zerrendako elementuak errenkada eta zutabearen arabera gorantz ordenatuta daude:



Honako **azpiprograma hau inplementatu** behar da:

```
public Node {
   Integer balioa;
   Integer errenkada;
   Integer zutabea;
   Node    next;

   public Node(Integer pErrenk, Integer pZut, Integer pBalioa){
       errenkada = pErrenk; zutabea = pZut; balioa = pBalioa; next = null;
   }
}

public class Matrize {

         Node first;

         public Matrize batura(Matrize m1, Matrize m2) {
         // aurre:
         // post: emaitza m1 eta m2 matrizeen batura da

}
```

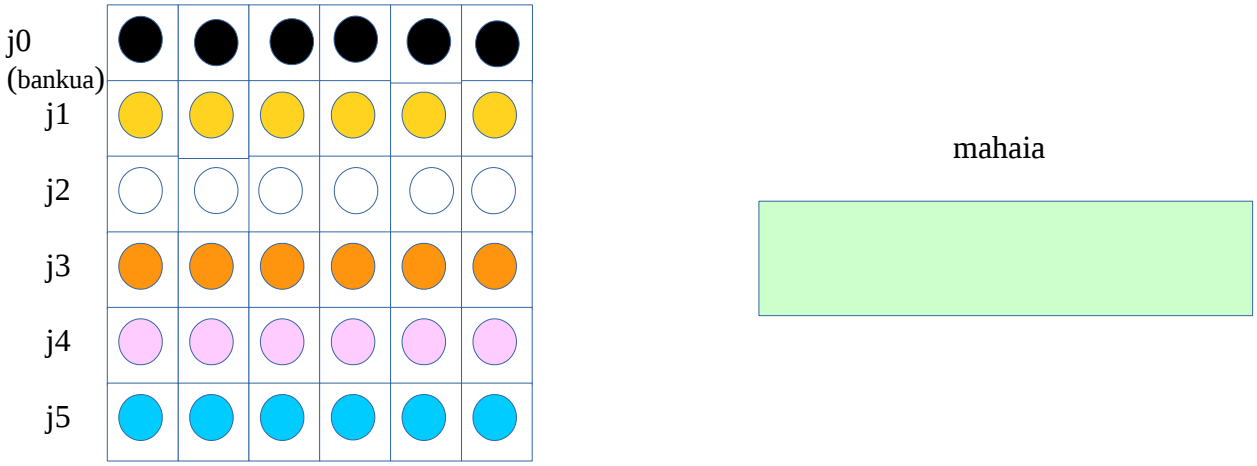
Erabilitako algoritmoaren **konplexutasun ordena O(N) izan behar du**, N balioa M1 eta M2 matrizeetan 0 ez diren elementuen batezbesteko kopurua izanda.

Adibidez:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} + \begin{pmatrix} 9 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix} = \begin{pmatrix} 9 & 0 & 0 & 0 \\ 0 & 10 & 0 & 5 \\ 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 5 \end{pmatrix}$$

2. Koloreen jokoa (1,5 puntu)

Joko bateko 5 jokalarik kolore bateko fitxa-kopuru bat dute (jokalari guztiek hasten dute jokoa kopuru berarekin). Jokalari bakoitzak hasieran kolore bereko fitxak izango ditu, eta jokalari guztiek kolore desberdinak dituzte. Gainera, bankuak kolore beltzeko fitxak ditu (bankua zerogarren jokalaria izango da).Jokoak mahai bat behar du, bertan fitxak ondoren azaltzen diren arauen arabera jarriko direlarik. Irudi honek jokoaren hasierako egoera erakusten du:



Jokaldi bakoitzen bi dado botako dira, ondoko arauekin:

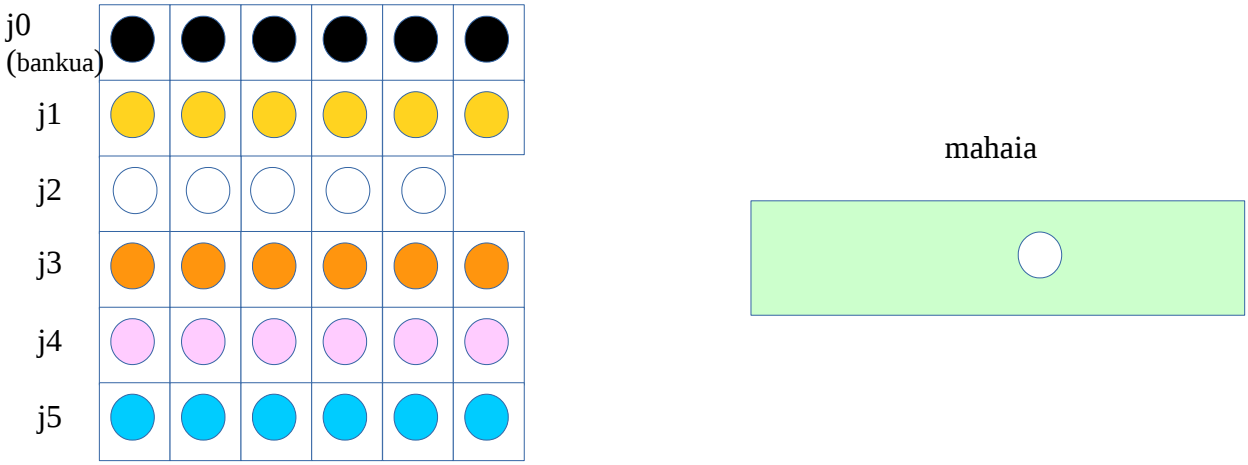
1. Lehen dadoaren balioa 6 baldin bada, jokoa amaitzen da.

2. Bestela, lehen dadoa bikoitia baldin bada, bigarren dadoaren balioak esango du zein jokalarik mugitzen duen fitxa bat mahaira. Dadoaren balia 1etik 6koa denez, balio horri bat kenduko zaio [0, 5] tarteko balioa lortzeko. Jokalariak fitxarik ez baleuka, ez da ezer egingo.

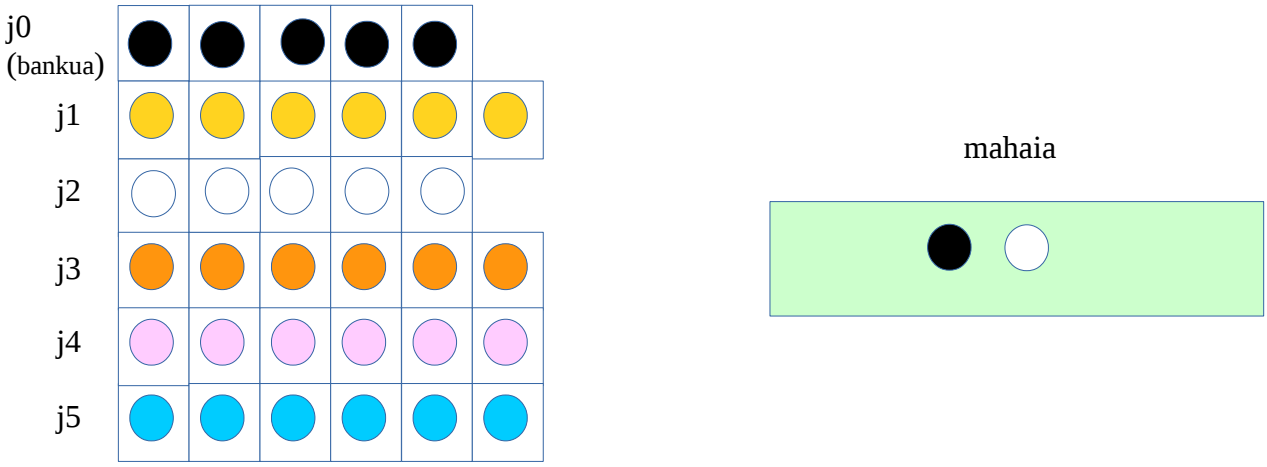
3. Lehen dadoa bakoitia baldin bada, fitxa bat mugituko da mahaitik bigarren dadoaren balioak esango duen jokalarira. Fitxa hori jokalariak erabiliko duen azkena izango da. Mahaian fitxarik ez balego, ez da mugimendurik egingo.

4. Irabazlea fitxa beltz gehien duen jokalaria izango da (bankua kontuan hartu gabe). Berdinketaren kasuan, irabazlea puntuazio maximoa duen lehen jokalaria izango da.

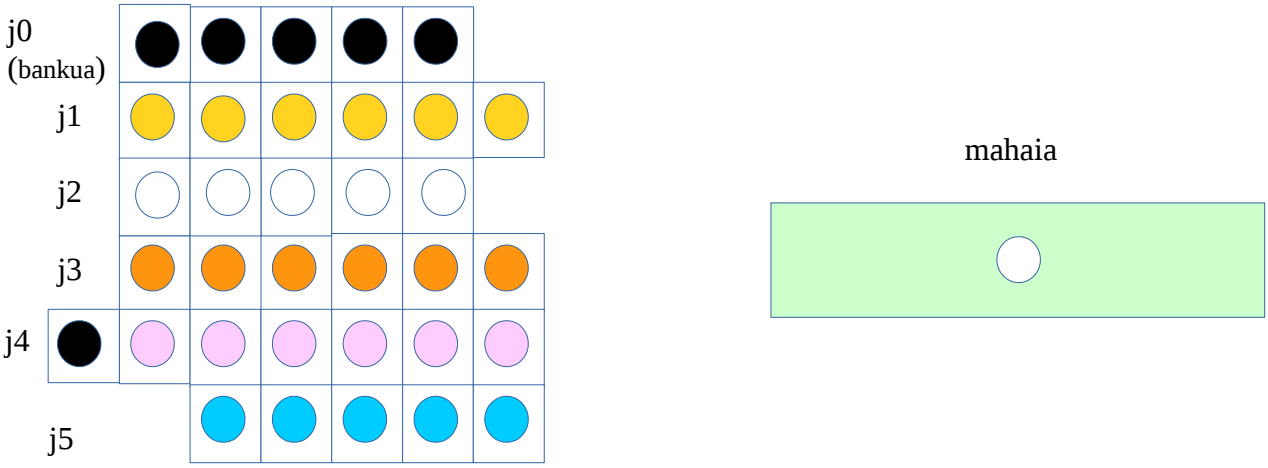
Adibidez, hasierako egoeran dadoek (4, 3) emango balute, bigarren jokalariaren fitxa bat hartuko litzateke (3 ken 1) eta mahaian jarri beharko litzateke:



Dadoak berriro botatzen direnean (4, 1) bikotea aterako balitz, fitxa bat mugituko da 0garren jokalaritik mahaira:



Eta orain (3, 5) aterako balitz, lehenengo dado bakoitia denez, fitxa bat mugituko da mahaitik 4garren jokalarira. Kontuan izan mugituko den fitxa mahaian kokatu zen azkenekoa izango dela. Gainera, fitxa hori 4garren jokalariak jokatuko duen azkena izango da.

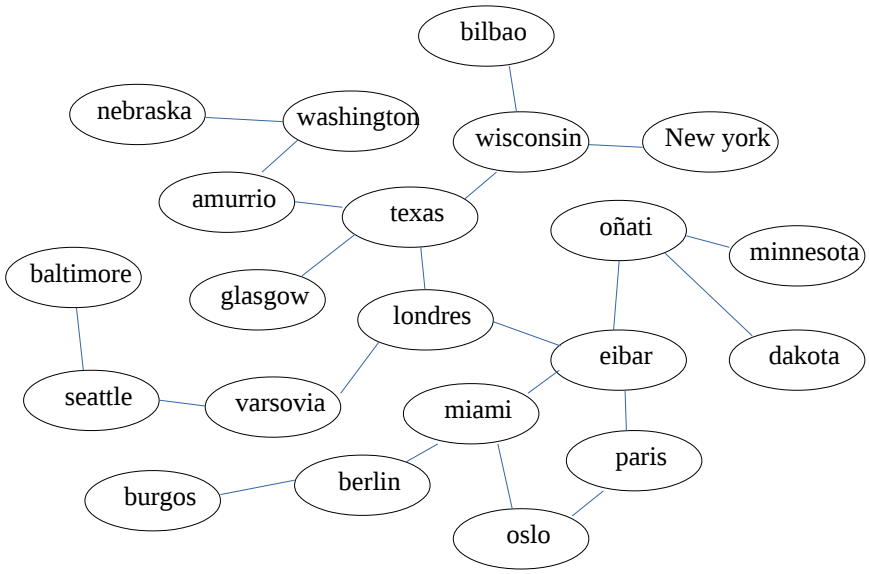


Azpiprograma hau inplementatu behar da:

```
public class Jokoa {

         Queue<Integer>[] jokalariak;
         // Fitxen koloreak balio osokoen bidez adierazten dira: beltzak 0 eta
         //  beste jokalarien kolorea bere posizioarekin bat etorriko da (hau da,
         //  1 jokalariak 1 kolorea, ...)

         Stack<Integer> mahaia;

         public int jokoa(int n, ArrayList<Jokaldi> jokaldiak) {
         // aurre: n jokalari bakoitzaren hasierako fitxa-kopurua da
         //      "jokaldiak" zerrendak partida bateko jokaldiak ditu
         // post: emaitza irabazlearen zenbakia da
}
public class Jokaldi {
   int dado1;
   int dado2;
}
```

3. Lurrikara (1,5 puntu)

Irudiko grafo ez zuzenduak lurraldeak adierazten ditu. Bi lurralderen arteko arku batek lurraldeek elkarrekin muga dutela adierazten du.



Lurralde batean N intentsitateko lurrikara bat gertatzen denean, lurrikara alboko lurraldeetara mugitzen da Nren intentsitate erdiarekin, eta era horretan indarra galtzen doa.

Jakin nahi da zeintzuk diren N intentsitateko lurrikara batek kaltetutako lurraldeak. Lurralde batean lurrikarak eragina izango du intentsitatea 1 edo altuagoa baldin bada.

Adibidez, Londresen 6 intentsitateko lurrikara bat balego, *texas*, *eibar* eta *varsovia* kaltetuko lituzke 3 indarreko intentsitatearekin, eta *glasgow*, *amurrio*, *wisconsin*, *oñati*, *paris*, *miami*, eta *seattle* 1,5 intentsitatearekin.

```
public class Graph
{
   protected final int DEFAULT_CAPACITY = 100;
   protected int numVertices;   // number of vertices in the graph
   protected boolean[][] adjMatrix;   // adjacency matrix
   protected String[] vertices;   // values of vertices

   public int index(String t) { // EZ DA INPLEMENTATU BEHAR!
   // aurre: elementua grafoan dago
   // post “vertices” arrayan t-ren posizioa emango du

   public ArrayList<String> kaltetutakoak(int intentsitatea, String l)
   // aurre:  “intentsitatea” parametroak lurrikararen indarra ematen du
   //       “l” lurrikara gertatu den lurraldea
   // post: emaitza kaltetutako lurraldeen zerrenda da
   //    Lurralde bat kaltetua izango da lurrikararen intentsitatea 1 edo altuagoa
   //      baldin bada.
   //    Lurrikararen intentsitatea erdira jaitsiko da lurralde batetik ondokoetara
   //      pasatzen denean.
}
```

Hau eskatzen da:

1. “**`kaltetutakoak`**(...)” metodoaren inplementazioa
2. Kalkulatu, modu arrazoituan, algoritmoaren kostua.

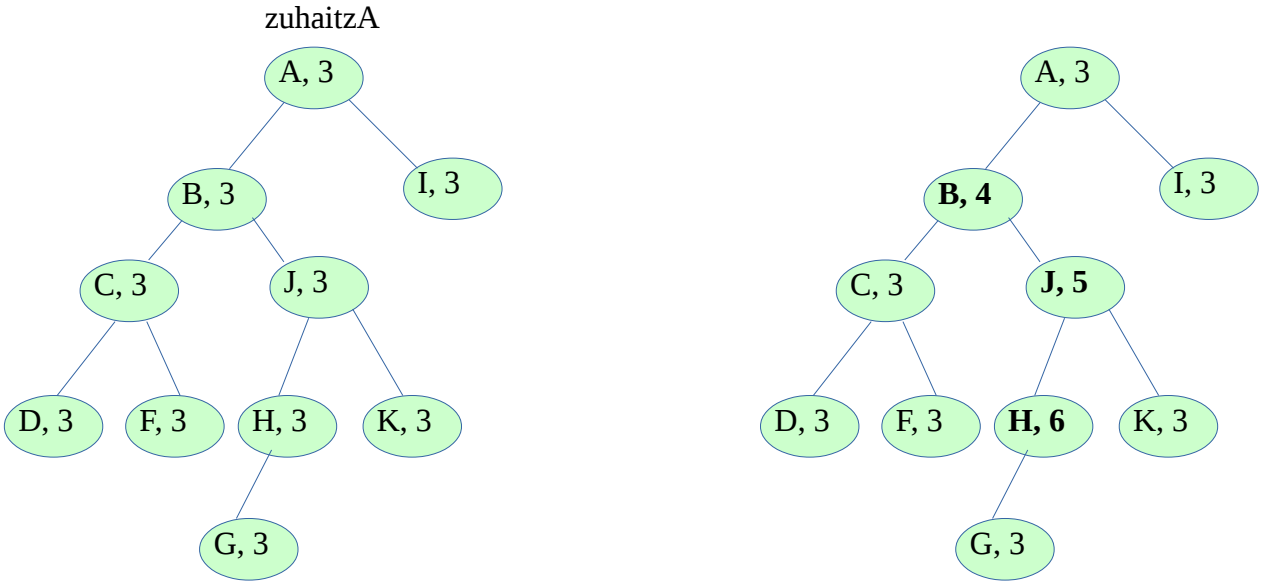
4. Jokoaren zuhaitza (1,5 puntu)

Joko bateko informazioa gordetzeko zuhaitz bitar bat dugu. Jokalari bakoitzak puntuazio bat dauka.

```
public class Info {
    String   s;
    Integer  puntuak;
}

public class Node {
    Info     content;
    Node     ezkerra, eskuina;
    Node     gurasoa;
}

public class Zuhaitza {
   private Node root;

   public void saritu(int puntuak, String elem);
}
```

Jokalari batek puntuak lortzen dituenean, jokoak esaten du jokalari horri n puntu emango zaizkiola, eta puntuak banatuko direla bere arbasoen artean: (n-1) puntu bere gurasoari, (n-2) hurrengoari eta abar.

Adibidez, (a) irudiko zuhaitzean “zuhaitzA.saritu(3, “H”)” egingo bagenu, (b) irudiko zuhaitza emango luke, H-ri 3 puntu, J-ri 2 puntu eta B-ri puntu bat emanez.



(a) (b)

Hau eskatzen da:

1. “**`saritu`**(...)” metodoaren inplementazioa

2. Kalkulatu, modu arrazoituan, algoritmoaren kostua.