

## 4. eginkizuna

Bi funtzionalitate berri inplementatu nahi ditugu:

```
a) public HashMap<String, Double> pageRank()  
//POST: emaitza aktore edo pelikula bakoitzaren PageRank algoritmoaren  
balioa da
```

PageRank algoritmoa (<https://eu.wikipedia.org/wiki/PageRank>) grafo baten dokumentuei (edo web-orriei) bere garrantzia adierazten duen zenbakizko balio bat esleitzen dien algoritmo bat da. PageRank sistema Google bilatzaileak erabiltzen du web-orri baten garrantzia erabakitzeko. PageRank-ek pertsona batek grafoa atzi eta estekak jarraituz nodo zehatz batetara iristeko duen probabilitatea kalkulatu du. PageRank algoritmoak zenbait iterazioen beharra dauka, balio hurbildua doitzeko. Hasiera batean, nodo guztiek probabilitatea berbera dute:

$\forall A \in \text{nodoak}, \quad PR(A) = \frac{1}{N}$ , non N grafoaren nodo kopurua den.

Gero, iterazio bakoitzean, nodo bakoitzaren balioa birkalkulatu behar da honako formula erabiliz:

$$PR(A) = \frac{1-d}{N} + d * \sum_{i=1}^n \frac{PR(i)}{C(i)} \quad \text{non}$$

- **PR(A)** A orriaren PageRank da.
- **d** (damping factor) indargetze faktorea da, 0 eta 1 arteko balioa duena (balio tipiko bat 0.85 da)
- **PR(i)** A estekatzen duen i orri bakoitzaren PageRank balioa (aurreko iteraziokoa).
- **C(i)** i orritik ateratzen diren esteka kopurua (bai A orrialdera, bai beste orrialdeetara).

Prozesuak jarraituko du iterazio baten eta hurrengo iterazioaren kenketen balio absolutuen batuketa aurredefinitutako atalase bat baino txikiagoa izan arte (adibidez 0.0001).

Adibidez, demagun lau nodoez osatutako grafoa dugula : A, B, C eta D nodoak. B nodoak estekak ditu C eta A nodoetara, C nodoak esteka bat du A nodora, eta D nodoak estekak ditu A, B eta C nodoetara.

Hasieran nodo bakoitzaren balioa 0.25 izango da. Lehenengo iterazioan, B nodoak bere balioaren erdia, 0.125, pasatuko dio A nodoari, eta beste erdia C nodoari. C nodoak bere balio osoa, 0.25, pasako dio A nodoari. D nodoak hiru esteka dituenenez, bere balioaren herena (gutxi gorabehera 0.083) pasako die A, B eta C nodoei. Formula aplikatu ostean, lehenengo iterazioa bukatzerakoan, A nodoaren PageRank balioa 0.427 izango da gutxi gorabehera.

```

b) public class Bikote {

    String aktoreaEdoPelikula;

    Double pagerank;

}

public ArrayList<Bikote> bilatzailea(String gakoHitz)

// Post: Emaidza emandako gako-hitzarekin lotuta dauden elementuen
zerrenda da (aktoreak edo pelikulak, gakoaren arabera), bere pagerank-
aren arabera handienetik txikienera ordenatuta (hau da, lehenengo
posizioetan pagerank handiena duten elementuak agertuko dira).

Adibidez, emandako katea aktore bat baldin bada, orduan metodo honek
bere pelikulen zerrenda ordenatua bueltatuko du, pagerank zenbakiaren
arabera (eta, gakoa pelikula bat baldin bada, orduan pelikula horretako
aktoreak lortuko dira, pagerank-aren arabera ordenatuta).

```

Honakoa entregatuko da:

- Eskatutakoa exekutatzen duten programak (zuzen exekutatu behar dira). **Frogatu egin beharko da programak ondo funtzionatzen duela datu ez tribialekin (hau da, hasierako fitxategiko milaka lerro prozesatuta).**
- Eskatutako metodoen exekuzio-adibideak, proba-datuak eta emaitzak nolakoak diren azalpena. Interesgarria denean, proba bakoitzaren exekuzio denbora ere emango da.
- Dokumentazioa, emandako problema, aztertutako aukerak, inplementazioa, eraginkortasuna etabar deskribatzen duena.

Gainera Checklist-a bete eta entregatu beharko duzue, eskatutako guztia egin duzuela egiaztatzeko.

**OHARRA:** eskatutako emaitza batzuek konputazio-kostu altua dutenez, horrek algoritmoak entregatze-data baino lehenago funtzionatzea eskatuko du, bestela ezin izango direlako emaitzak entregatu.