

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1180

**FACEBOOK APLIKACIJA ZA MJERENJE
POPULARNOSTI**

Barbara Carević

Zagreb, svibanj 2009.

Sadržaj

1.Uvod.....	5
2.Algoritmi	6
2.1.PageRank.....	6
2.1.1.Algoritam	7
2.1.2.Pojednostavljeni algoritam.....	7
2.1.3.Faktor prigušenja.....	9
2.1.4.Matrica hiperlinkova	11
2.1.5.Računanje vektora I	13
2.1.6.Tri bitna pitanja	14
2.1.7.Probabilistička interpretacija matrice H	15
2.1.8.Završna modifikacija.....	19
2.2.Najkraći put.....	20
2.2.1.Dijkstrin najkraći put.....	20
2.2.2.Algoritam	20
2.2.3.Primjer.....	21
3.Facebook	25
3.1.Facebook platforma.....	25
3.1.1.API	25
3.1.2.FBML.....	26
3.1.3.FQL	27
3.2.Točke integracije	28
3.3. Postavljanje Facebook aplikacije	31
4.Implementacija	37
4.1. Arhitektura	38
4.2.Baza.....	40
4.3.Izgled i sučelja aplikacija	42

4.3.1. Grafičko sučelje aplikacije Six Degrees to Barbara.....	42
4.3.2.Sučelje aplikacije PageRankCalc	47
4. Diskusija.....	49
6.Zaključak	51
7.Literatura	52

1.Uvod

U današnje vrijeme internetske društvene mreže privlače jako veliki broj korisnika, koji svojom aktivnošću stvaraju ogromnu količinu podataka. Zbog toga su postale predmet proučavanja raznih grana znanosti, među kojima su sociologija, antropologija, biologija, te računalne znanosti. Jedna od trenutno najpopularnijih računalnih mreža je Facebook, s preko 200,000,000 registriranih korisnika.

Proučavanje velikih količina povezanih podataka je čest problem u računalnoj znanosti, a pogotovo području pretraživanja web stranica. Najpoznatiji algoritam za to je PageRank[1] kojeg su osmislili Larry Page i Sergey Brin. Njih dvojica su kasnije osnovali Google, danas uvjerljivo najpopularniji internet pretraživač, na čije rezultate pretrage, među ostalim faktorima, utječe i PageRank.

U ovom radu pokušalo se na Facebook mrežu primijeniti PageRank algoritam za mjerenje popularnosti korisnika. Uz to računala se i udaljenost svih korisnika od početnog člana u mreži. U tu svrhu razvijena je Facebook aplikacija „Six Degrees to Barbara“.

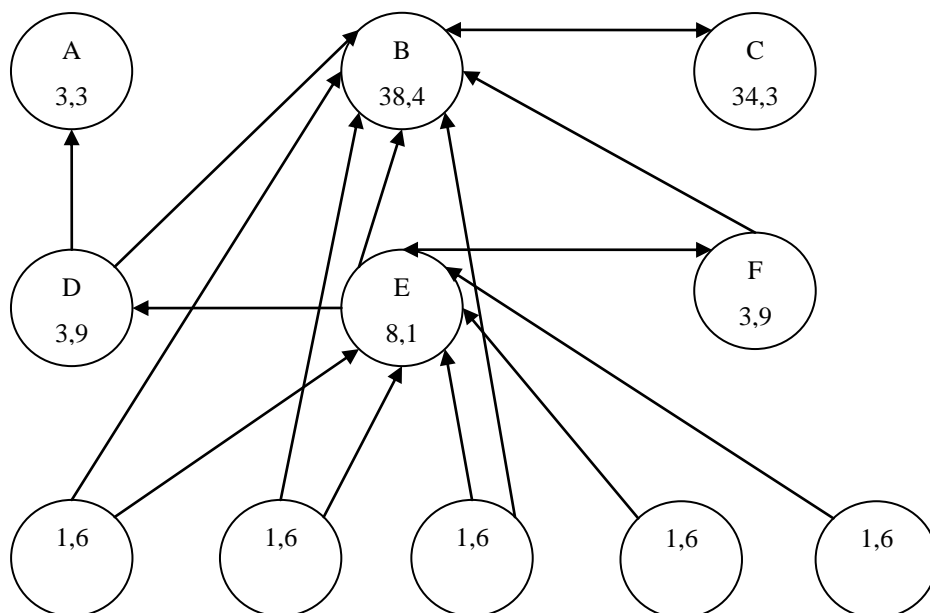
Korišteni algoritmi su opisani u poglavlju "2. Algoritmi". Arhitektura aplikacije i Facebook platforma poglavlju "3. Implementacija". Diskusija je dana u poglavlju "6. Diskusija", a sažetak rada i zaključak u poglavljima "7. Sažetak" i "8. Zaključak". Popis korištene literature nalazi se u poglavlju "9. Literatura".

2.Algoritmi

2.1.PageRank

PageRank algoritam je trenutno najpopularnija metoda za vrednovanje dokumenata međusobno povezanih u određenu strukturu. Najčešće se koristi za web stranice. Svakom elementu se pridaje specifična brojčana vrijednost koja predstavlja „važnost“ elementa u strukturi. Brojčana vrijednost koja se pridaje elementu E se naziva PageRank od R i označava se s $PR(E)$.

Osnovni koncept algoritma je da PageRank elementa raste s brojem drugih elemenata koji pokazuju na njega. PageRank elementa se računa pomoću PageRankova elemenata koji pokazuju na njega, a njihov PageRank pomoću PageRankova elemenata koji pokazuju na njih. Dakle, PageRank elementa se uvijek računa rekurzivno pomoću PageRanka ostalih stranica. Prema tome, PageRank svakog elementa utječe na PageRank svih ostalih elemenata, te zapravo ovisi o cijelom grafu. Na sljedećoj slici su prikazane vrijednosti PageRanka za jednostavnu mrežu.



Slika 1: Primjer PageRank vrijednosti[2]

Stranica C ima viši PageRank od stranice E, iako ima manje veza koje vode do nje. Veza koju ima sa stranice B se puno više vrednuje, jer stranica B ima visoki PageRank zbog puno dolaznih veza.

Algoritam su razvili i objavili Sergey Brin i Larry Page [*S. Brin, L. Page, 'The PageRank Citation Ranking: Bringing Order to the Web'; 'The Anatomy of Large-Scale Hypertextual Web Search Engine'*] krajem devedesetih na sveučilištu Stanford u sklopu istraživačkog projekta : *Internet tražilice nove generacije*. Ime „PageRank“ je zaštićeno od strane Googlea, a algoritam je zaštićen patentom od strane sveučilišta Stanford. Googleu su dodijeljena ekskluzivna prava na korištenje patenta.

2.1.1.Algoritam

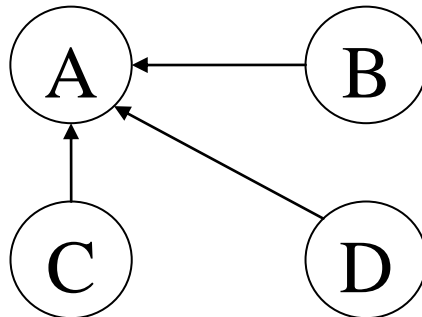
PageRank je raspodjela vjerojatnosti koja predstavlja vjerojatnost da će osoba nasumičnim praćenjem izlaznih veza doći na određenu stranicu. PageRank se može izračunati za skup dokumenata bilo koje veličine. Pretpostavlja se da je raspodjela vjerojatnosti jednoliko podijeljena na početku procesa računanja. Računanje zahtjeva više iteracija kroz skup dokumenata da bi aproksimirane vrijednosti PageRanka bile bliže teoretskoj točnoj vrijednosti.

Vjerojatnost se predstavlja kao brojčana vrijednost između 0 i 1, prema tome vrijednost PageRanka 0.5 govori da postoji 50% šanse da će surfer nasumično prateći izlazne veze doći do stranice s PageRankom 0.5.

2.1.2.Pojednostavljeni algoritam

Pretpostavimo sustav od četiri web stranice: A, B, C i D. Početna aproksimacija PageRank bila bi jednoliko podijeljena između ta četiri elementa. Prema tome, svaki element bi dobio procijenjeni PageRank 0,25.

Ako stranice B, C i D pokazuju samo na A, onda bi svaka od njih pridonijela s 0,25 PageRanku od A. Ukupan PageRank ovog sustava bi bio dodijeljen stranici A, budući da sve veze pokazuju na A.

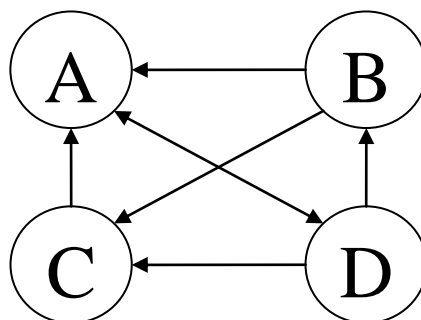


Slika 2: Primjer mreže u kojoj sve stranice pokazuju na jednu stranicu

$$PR(A) = PR(B) + PR(C) + PR(D)$$

Rezultat je 0,75.

Pretpostavimo da stranica B pokazuje i na stranicu C, a stranica D ima veze na sve tri ostale stranice. Vrijednost veza je podijeljena između svih izlaznih veza stranice. Stranica B daje vrijednost 0,125 stranici A i vrijednost 0,125 stranici C. Samo trećina PageRanka stranice C se uzima za PageRank stranice A.



Slika 3: Primjer mreže

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}$$

Drugim riječima, PageRank dan s izlaznom vezom L je jednak vlastitom PageRanku elementa podijeljenim s normaliziranim brojem izlaznih veza.

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}$$

U općem slučaju vrijednost PageRanka se može izraziti za svaki element u kao:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)}$$

Tj. Vrijednost PageRanka za element u ovisi o vrijednostima PageRanka za svaku stranicu v iz skupa B_u (ovaj skup sadrži sve elemente koji pokazuju na element u), podijeljen s brojem $L(v)$ veza od elementa v .

2.1.3. Faktor prigušenja

U teoriji PageRanka pretpostavlja se da će nasumični surfer u jednom trenutku prestati pratiti izlazne veze. Vjerojatnost, u svakom koraku da će osoba nastaviti pratiti izlazne veze je faktor prigušenja d . U raznim istraživanjima su testirani različiti faktori prigušenja, ali se općenito uzima da je faktor prigušenja najbolje postaviti oko 0,85.

Faktor prigušenja se oduzima od 1 (u nekim varijacijama algoritma, rezultat se dijeli s brojem elemenata u skupu) i taj broj se dodaje umnošku faktora prigušenja i sumi dolaznih PageRank vrijednosti.

$$PR(A) = 1 - d + d \left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} \right)$$

Ili, gdje je N broj elemenata u skupu

$$PR(A) = \frac{1 - d}{N} + d \left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} \right)$$

PageRank svakog elementa se dobije većim dijelom od PageRanka ostalih elementa. Faktor prigušenja smanjuje dobivenu vrijednost.

Formula koristi model nasumičnog surfera kojemu postane dosadno nakon nekoliko praćenja izlaznih veza, te se prebacuje na nasumičnu stranicu. PageRankom stranice dobiva se šansa da će nasumični surfer završiti na stranici praćenjem izlaznih veza. To se može predstaviti kao Markovljevi lanac u kojem su stanja stranice, a prijelazi koji su svi jednako vjerojatni veze među stranicama.

Ako stranica nema izlaznih veza na druge stranice s njom završava proces nasumičnog praćenja izlaznih veza. To se rješava tako da se u tom slučaju nasumično odabere neka druga stranica, te se nastavlja se sa surfanjem.

Kod računanja PageRanka se uzima da stranice koje nemaju izlaznih veza pokazuju na sve ostale stranice u skupu. Njihov PageRank se jednoliko dijeli između svih ostalih stranica.

Prema tome, slijedi formula:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

Gdje su p_1, p_2, \dots, p_N sve stranice koje se uzimaju u obzir, $M(p_i)$ je skup stranica koje pokazuju na p_i , $L(p_j)$ je broj izlaznih veza stranice p_j , N je ukupan broj stranica.

Vrijednosti PageRanka su članovi dominantnog svojstvenog vektora u modificiranoj matrici susjedstva. Svojstveni vektor je

$$R = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix},$$

gdje je R rješenje jednakosti.

$$R = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1, p_1) & \ell(p_1, p_2) & \dots & \ell(p_1, p_N) \\ \ell(p_2, p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i, p_j) & \\ \ell(p_N, p_1) & \dots & & \ell(p_N, p_N) \end{bmatrix} R$$

Gdje je funkcija susjedstva $\ell(p_i, p_j)$ jednaka nuli ako stranica p_j ne pokazuje na stranicu p_i , a normalizirana za svaki j .

$$\sum_{i=1}^N \ell(p_i, p_j) = 1$$

Tj. elementi svakog stupca sumirani daju 1.

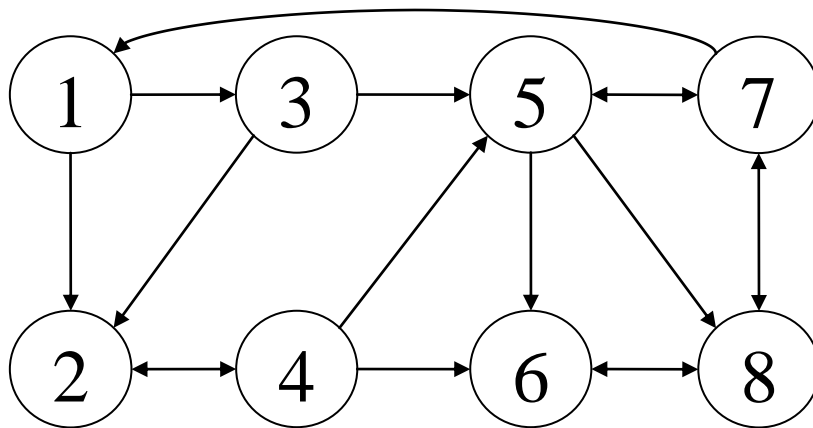
Zbog velikog prostora kojeg pokriva matrica susjedstva, vrijednosti PageRanka se brzo aproksimiraju. Kao rezultat Markovljeve teorije, može se pokazati da je PageRank stranice vjerojatnost da se dođe do stranice nasumičnim praćenjem izlaznih veza. Ta vjerojatnost je jednaka t^{-1} , gdje je t očekivani broja koraka do željene stranice.

2.1.4. Matrica hiperlinkova

Problem PageRanka se može prikazati na matematički pogodniji način[3]. Kreira se matrica, koja se naziva matrica hiperlinkova, $H = [H_{ij}]$ u kojoj su zapisi i-tog retku i j-tog stupca:

$$H_{ij} = \begin{cases} \frac{1}{l_j} & \text{ako } P_j \in B_i \\ 0 & \end{cases}$$

Matrica H ima neka osobita svojstva. Prvo, svi zapisi su nenegativni. Također, suma svih zapisa u stupcu je jednaka 1, osim ako stranica koja odgovara tom stupcu nema veza. Matrice u kojima su svi zapisi nenegativni i u kojima je suma svih zapisa u svakom stupcu jednaka jedan nazivaju se stohastičke matrice. Formira se vektor $I = [I(P_i)]$, čije su komponente vrijednosti PageRanka svih stranica. Uvjet koji definira PageRank može se izraziti kao $I = HI$. Drugim riječima, vektor I je svojstveni vektor matrice H sa svojstvenom vrijednosti 1. Taj vektor se također naziva i stacionarni vektor od H . U primjeru je dan prikaz malog skupa od osam web stranica, veze su označene strelicama.



Slika 4: Primjer mreže od osam stranica

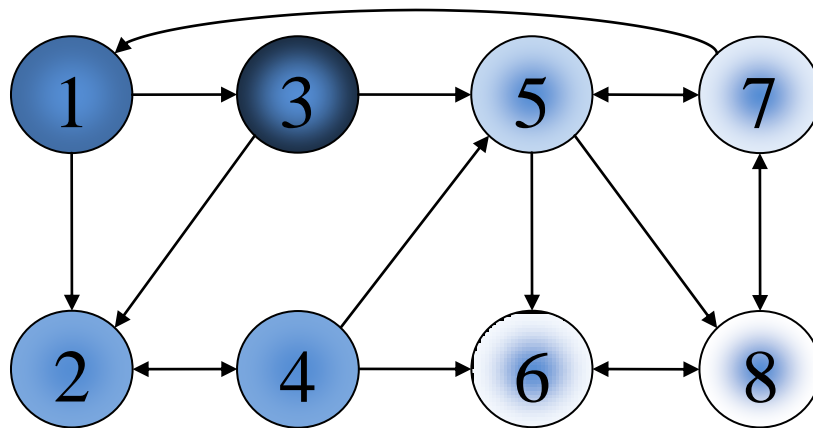
Odgovarajuća matrica je $H =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 1/2 & 0 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 1 & 1/3 & 0 \end{bmatrix},$$

sa stacionarnim vektorom $I =$

$$\begin{bmatrix} 0,0600 \\ 0,0675 \\ 0,0300 \\ 0,0675 \\ 0,0975 \\ 0,2025 \\ 0,1800 \\ 0,2950 \end{bmatrix}$$

Stranica 8 ima najveću vrijednost PageRanka. Na sljedećoj slici je prikazan je isti graf, ali tako da su stranice s većom vrijednosti PageRanka obojane svjetlije.



Slika 5: Prikaz vrijednosti PageRanka bojom

2.1.5. Računanje vektora I

Postoji puno načina za računanje svojstvenih vektora kvadratne matrice. Međutim, u ovom slučaju uzima se u obzir veličina matrice H , svaki stupac odgovara web stranici koju je indeksirao Google. To znači da matrica ima oko 25 milijardi stupaca i redaka, iako su većina zapisa u H jednaki nuli. Ta činjenica se koristi u „power“ metodi.

Prvo se odabire vektor I^0 kao kandidat za I , a zatim se stvara niz vektora I^k kao

$$I^{k+1} = HI^{k+1}$$

Osnovni princip: Niz I^k će konvergirati u stacionarni vektor I

To se može ilustrirati na prethodnom primjeru s osam stranica.

Tablica 1: Računanje PageRanka za primjer s osam stranica

I^0	I^1	I^2	I^3	...	I^{60}	I^{61}
1	0	0	0	...	0,06	0,06
0	0,5	0,25	0,1667	...	0,0675	0,0675
0	0,5	0	0	...	0,03	0,03
0	0	0,5	0,25	...	0,0675	0,0675
0	0	0,25	0,1667	...	0,0975	0,0975
0	0	0	0,25	...	0,2025	0,2025
0	0	0	0,0833	...	0,18	0,18
0	0	0	0,0833	...	0,295	0,295

Ne postoji apsolutna mjera važnosti stranice, samo relativne mjere koje uspoređuju važnosti dvaju stranica kroz izjave kao što su „Stranica A je dvaput važnija od stranice B“ iz kojih dolazimo do vrijednosnog poretka. Iz ovog razloga, sve mjere važnosti se mogu množiti s nekom fiksnom vrijednosti bez da se promijeni informacija koju donose. Na ovaj način, uvijek će se pretpostavljati, iz razloga koji će brzo biti objašnjeni da je suma svih vrijednosti važnosti jednaka jedan.

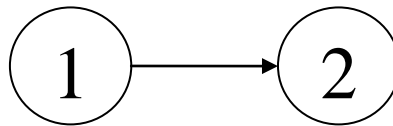
2.1.6. Tri bitna pitanja

Postavljaju se tri bitna pitanja:

- Konvergira li uvijek niz I^k ?
- Je li vektor prema kome konvergira nezavisan od vektora I^0 ?
- Sadrži li vrijednosni poredak sve informacije koje su potrebne?

Gleda li se dosad korištena metoda, odgovor niti na jedno pitanje nije potvrđan. Međutim, metoda se može modificirati da svi odgovori budu „da“.

Uzima se jednostavan primjer male mreže koja se sastoji samo od dvije stranice, od kojih jedna pokazuje na drugu.



Slika 6: Primjer jednostavne mreže

Odgovarajuća matrica je $H = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$.

Ovo je jedan od načina na koji algoritam može nastaviti:

Tablica 2: Računanje PageRanka za primjer s dvije stranice

\mathbf{I}^0	\mathbf{I}^1	\mathbf{I}^2	$\mathbf{I}^3 = \mathbf{I}$
1	0	0	0
0	1	0	0

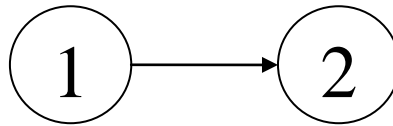
U ovom slučaju vrijednosni poredak obje stranice je nula, što ništa ne govori o važnosti stranica. Problem je što stranica 2 nema izlaznih veza. Važnost stranice 1 se uzima, ali se ne prenosi na druge stranice. Rezultat je otjecanje cijele vrijednosti važnosti iz mreže. Stranice bez izlaznih veza nazivaju se viseće (eng. dangling) stranice. Za rješavanje ovog problema potrebno je modificirati matricu H i stacionarni vektor I .

2.1.7. Probabilistička interpretacija matrice H

Potrebno je drugačije definirati koncept nasumičnog surfera. Dosad je logika bila da ako je surfer na stranici P_j s l_j izlaznih veza, od kojih jedna vodi do stranice P_i , vjerojatnost da će u sljedećem koraku završiti na stranici P_i je $1/l_j$. Sada se uvodi vremenska komponenta. Dio vremena T_j označava vrijeme koje se provede na stranici P_j , prema tome dio vremena koji će se provesti na stranici P_i dolazeći od stranice P_j je jednak T_j/l_j jer ako je surfer završio na stranici P_i morao je doći sa stranice koja ima izlaznu vezu na nju. To znači

$$T_i = \sum_{P_j \in B_i} T_j / l_j$$

gdje je suma izražena preko svih stranica P_j koje imaju izlaznu vezu na P_i . Može se uočiti da ista jednakost definira vrijednosti PageRanka $I(P_i) = T_i$. Na taj način PageRank stranice se definira kao dio vremena kojeg će nasumični surfer potrošiti na stranici. Ovo se može intuitivno shvatiti kao surfanje o nepoznatoj temi, ako se neko vrijeme prate izlazne veze, na neke će se stranice nailaziti češće nego na druge. Te stranice su obično važnije. S ovom interpretacijom, prirodno je tražiti da suma zapisa u PageRank vektoru I bude 1. Međutim, postoji komplikacija u ovom opisu, ako se nasumično surfa, u nekom trenutku zaglaviti će se na stranici bez izlaznih veza (viseća stranica). Da bi se nastavilo sa surfanjem, sljedeća stranica se bira nasumično, tj. prividno takva stranica ima izlazne veze na sve ostale stranice. Ovo ima za efekt modifikaciju matrice hiperlinkova H . Stupac s svim nulama, koji predstavlja stranicu bez izlaznih veza, zamjenjuje se sa stupcem u kojem su svi zapisi $1/n$. Nova matrica se naziva matrica S . Za prethodni primjer vrijedi.



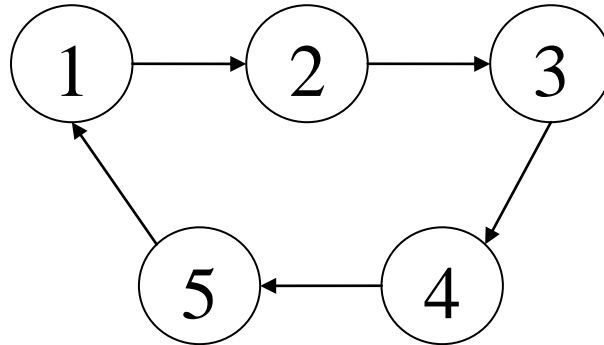
Slika 7: Primjer jednostavne mreže

Matrica $S = \begin{bmatrix} 0 & 1/2 \\ 1 & 1/2 \end{bmatrix}$ i svojstveni vektor $I = \begin{bmatrix} 1/3 \\ 2/3 \end{bmatrix}$.

Stranica 2 je dvaput važnija od stranice 1, što je i ispravno. Matrica S je također stohastička. Stohastičke matrice imaju nekoliko svojstava koja će se pokazati korisnim, npr. uvijek imaju stacionarne vektore. Matrica S se lako dobiva iz matrice H . Ako je A matrica u kojoj su svi zapisi nula osim stupaca koji odgovaraju stranicama bez izlaznih veza, u kojim su svi zapisi jednaki $1/n$, onda vrijedi $S = H + A$.

Općenito, *power* metoda je tehnika nalaženja svojstvenog vektora kvadratne matrice koji odgovara svojstvenoj vrijednosti s najvećom magnitudom. U ovom slučaju traži se svojstveni

vektor od S koji odgovara svojstvenoj vrijednosti 1. U najboljim okolnostima, ostali svojstveni vektori od S će imati magnitudu manju od 1, tj. $1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$. Prepostavi se sljedeća mreža:



Slika 8: Primjer mreže s pet stranica

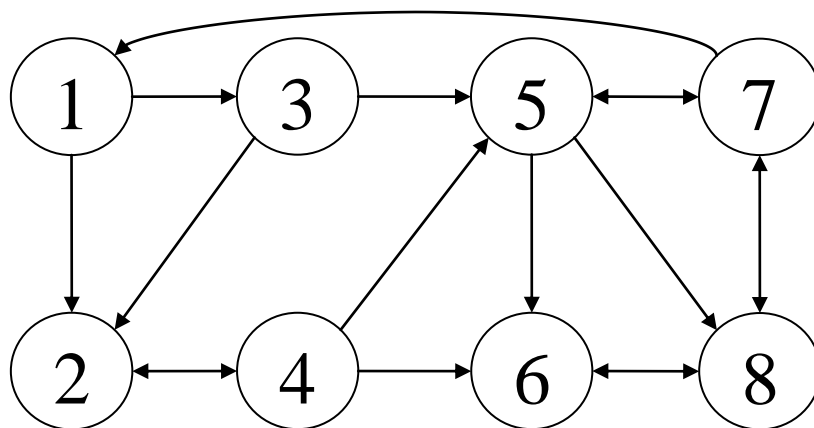
$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Algoritmom se dobiva:

Tablica 3: Računanje PageRanka za primjer s pet stranica

I^0	I^1	I^2	I^3	I^4	I^5
1	0	0	0	0	1
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0

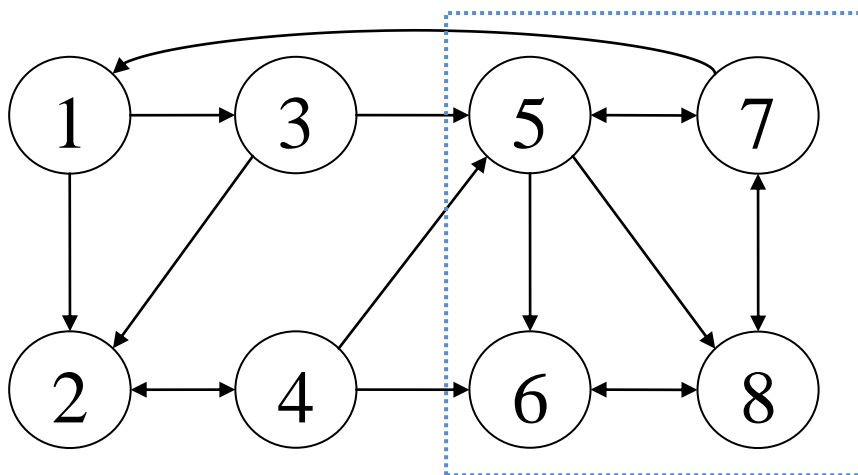
U ovom slučaju niz vektora I^k ne konvergira. Druga svojstvena vrijednost matrice S zadovoljava $|\lambda_2| = 1$ pa argument koji je dozvoljavao power metodu više ne stoji. Da bi se garantirao uvjet $|\lambda_2| < 1$, matrica S mora biti primitivna. To znači da za neke m , S^m ima sve pozitivne ulaze, tj. da je moguće doći od prve stranice do druge slijedeći m izlaznih veza. Očito, prethodni primjer ne zadovoljava ovo svojstvo. U sljedećem primjeru također dosadašnja metoda nije djelotvorna.



Slika 9: Primjer mreže s osam stranica

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 1 & 1/3 & 0 \end{bmatrix} \quad I = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0,12 \\ 0,24 \\ 0,24 \\ 0,4 \end{bmatrix}$$

Vrijednosti PageRanka za prve četiri stranice su nula, iako se to ne doima ispravnim jer sve četiri stranice imaju ulazne veze. Očito bi vrijednosni poredak svih stranica trebao biti pozitivan. Problem u ovom primjeru je što sadrži manju mrežu unutar sebe, kako je pokazano na slici ispod.



Slika 10: Prikaz mreže unutar veće mreže

U mrežu ulaze veze, ali ne izlaze. Događa se ista stvar kao u primjeru sa stranicama bez izlaznih veza, važnost se odvodi iz sustava. Ovo se događa kada je matrica S reducibilna, tj. može se zapisati na sljedeći način.

$$S = \begin{bmatrix} * & 0 \\ * & * \end{bmatrix}$$

Ako je matrica ireducibilna, može se garantirati postojanje stacionarnog vektora sa svim pozitivnim zapisima.

Mreža je snažno povezana ako za bilo koje dvije stranice, postoje veze koje vode od prve do druge. Takve mreže daju ireducibilne matrice S .

2.1.8. Završna modifikacija

Da bi se pronašla nova matrica, koja je i primitivna i ireducibilna, modificira se ponašanje nasumičnog surfera. Odabire se parametar α između 0 i 1. Surfer se s vjerojatnošću α vodi matricom S , s vjerojatnošću $1 - \alpha$ sljedeća stranica se odabire nasumično.

Ako se označi s 1 matricu $n \times n$ čiji su svi zapisi jednaki jedan, dobije se Google matrica:

$$G = \alpha S + (1 - \alpha) \frac{1}{n} 1$$

Budući da je G kombinacija stohastičkih matrica i sama je stohastička. Svi zapisi u G su pozitivni, pa je prema tome G primitivna i ireducibilna, te ima jedinstveni stacionarni vektor l koji se može koristiti za power metodu.

Uloga parametra α je vrlo bitna, ako je $\alpha = 1$, onda je $G = S$, što znači da se radi s originalnom strukturom mreže. Ako je $\alpha = 0$, onda je $G = 1/n_1$, tj. promatrana mreža ima izlaznu vezu između svake dvije stranice i gubi se originalna struktura mreže. Obično se uzima da je $\alpha = 0,85$.

2.2. Najkraći put

U teoriji grafova problem najkraćeg puta[4] je problem nalaženja puta između dva vrha (ili čvora) kojemu je suma težina svih sastavnih bridova minimalna. Primjer je nalaženje najkraćeg puta između dvije lokacije na karti. U ovom slučaju vrhovi predstavljaju lokacije, a bridovi segmente cesti i dana im je težina ovisno o vremenu koje treba da se pređu.

Formalno, ako je dan težinski graf (sa skupom od V vrhova i E bridova, $f: E \rightarrow R$), i elementom v iz V , pronalazi se put P iz v do v' tako da vrijedi da jr

$$\sum_{p \in P} f(p)$$

minimalna među svim putovima koji povezuju v i v' .

2.2.1. Dijkstrin najkraći put

Dijkstrin algoritam koji je osmislio nizozemski računalni znanstvenik Edsger Dijkstra 1959. godine je algoritam pretraživanja grafa koji rješava problem pronalaska najkraćeg puta iz jednog izvora za graf sa strogo pozitivnim vrijednostima bridova.

Dijkstra algoritam[5] je pohlepni algoritam koji gradi stablo najkraćih puteva. Zasniva se na pamćenju vrijednosti $d[v]$ trenutnog najkraćeg puta od s za svaki čvor v . U početku se dodjeljuju inicijalne vrijednosti udaljenosti, za početni čvor ta vrijednost iznosi 0, tj. $d[s] = 0$, a za ostale se uzima vrijednost beskonačno.

2.2.2. Algoritam

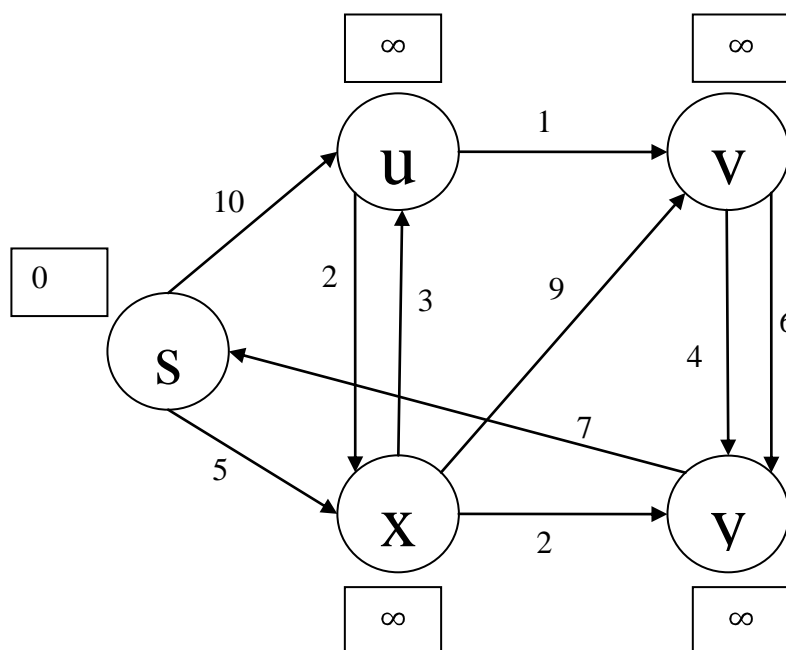
Nazovimo čvor s kojim počinjemo početni čvor, neka udaljenost od čvora X bude udaljenost od početnog čvora do X . Algoritam će dodijeliti početne vrijednosti udaljenosti i poboljšavati ih u sljedećim koracima.

1. Dodijeli se svakom čvoru vrijednost udaljenosti. Postavi se na nulu za inicijalni čvor i na beskonačno za ostale čvorove.
2. Označe se svi čvorovi kao neposjećeni. Postavi se inicijalni čvor kao trenutni.

3. Za trenutni čvor provjere se svi neposjećeni čvorovi i izračuna se njihova udaljenost (od inicijalnog čvora). Npr. ako trenutni čvor (A) ima udaljenost 6, i brid koji ga povezuje s drugim čvorom (B) je 2, onda je udaljenost do B preko A $6+2=8$. Ako je ova udaljenost manja od prethodno zabilježene udaljenosti (beskonačno u početku, nula za inicijalni čvor), onda se ta udaljenost uzima kao nova najkraća udaljenost.
4. Kad se provjere svi susjedni čvorovi trenutnog čvora, trenutni čvor se označi kao posjećen. Posjećeni čvor neće se ponovo provjeravati, njegova zabilježena udaljenost je konačna i minimalna.
5. Postavi se neposjećeni čvor s najmanjom udaljenosti (od inicijalnog čvora) kao sljedeći trenutni čvor i nastavlja se od koraka 3.

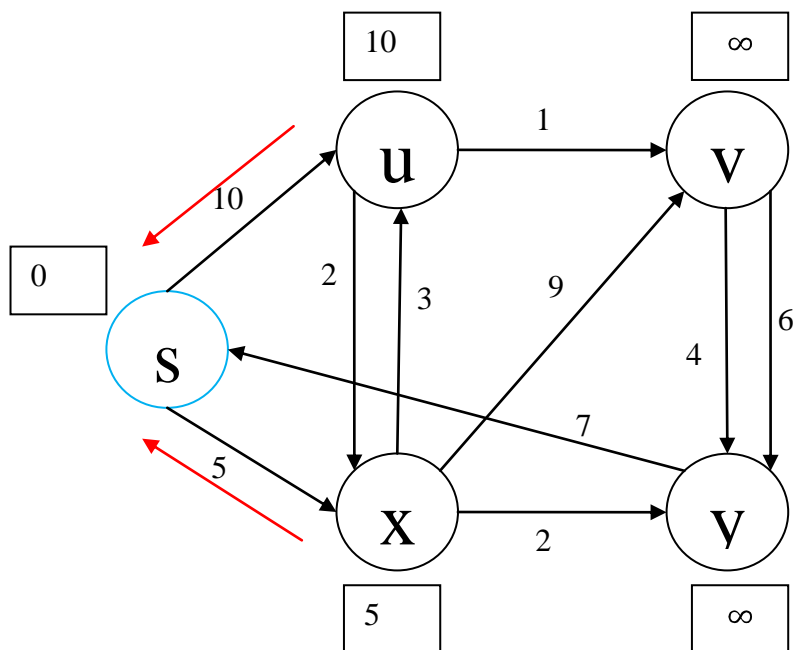
2.2.3.Primjer

Dijagramima je ilustriran Dijkstrin algoritam. U kvadratima su označene trenutne udaljenosti od početnog vrha.



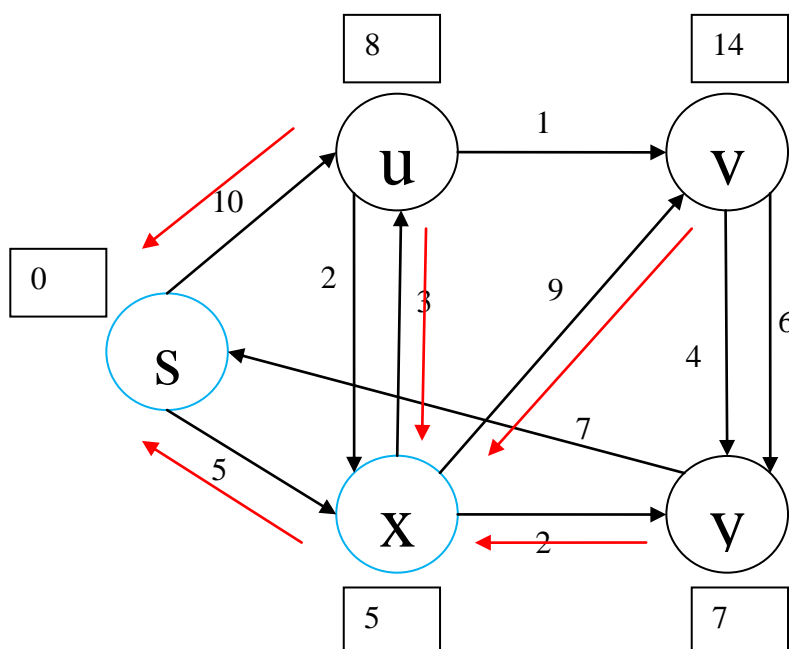
Slika 11: 1. korak algoritma

Prikazan je inicijalni graf. Svi čvorovi imaju udaljenost postavljenu na beskonačno osim početnog čvora.



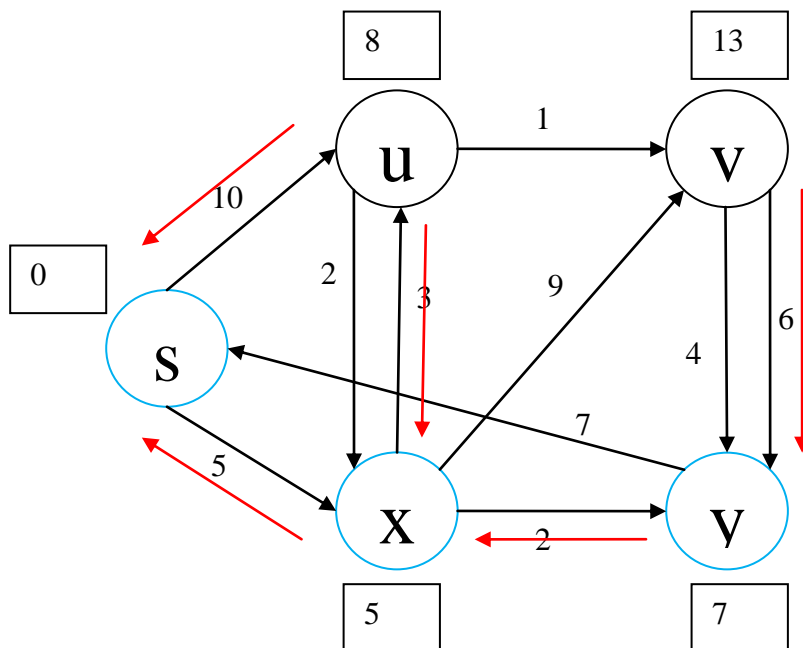
Slika 12: 2. korak algoritma

Izabere se čvor najbliži čvoru s . Dodaje se $d[s]$ vrijednosti S . Otpuste se svi čvorovi koji su susjedi čvoru s . Promijeni se lista prethodnika (crvene strelice) za sve čvorove kojima su prethodnici mijenjani.



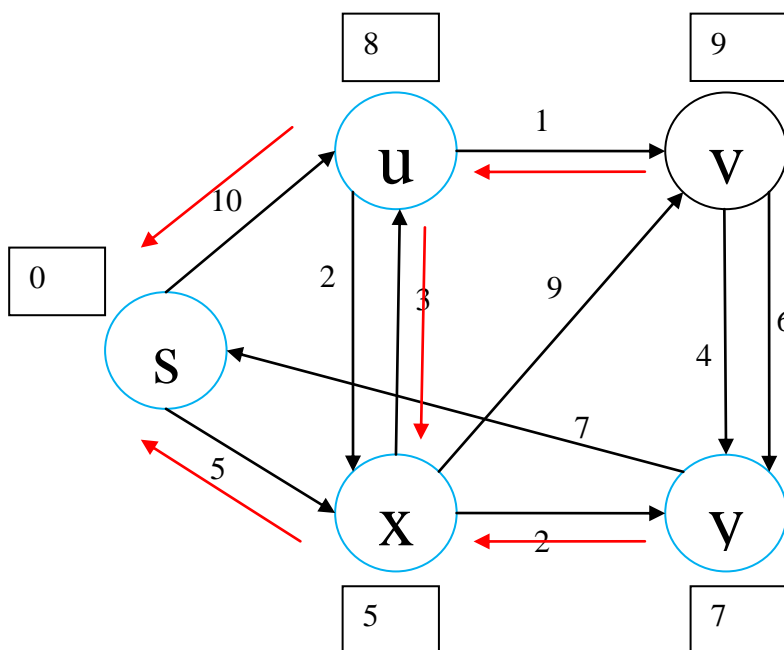
Slika 13: 3. korak algoritma

Odobere se najbliži čvor – x . Otpuste se svi čvorovi koji su susjedni čvoru x . Promijeni se lista prethodnika za u , v i y .



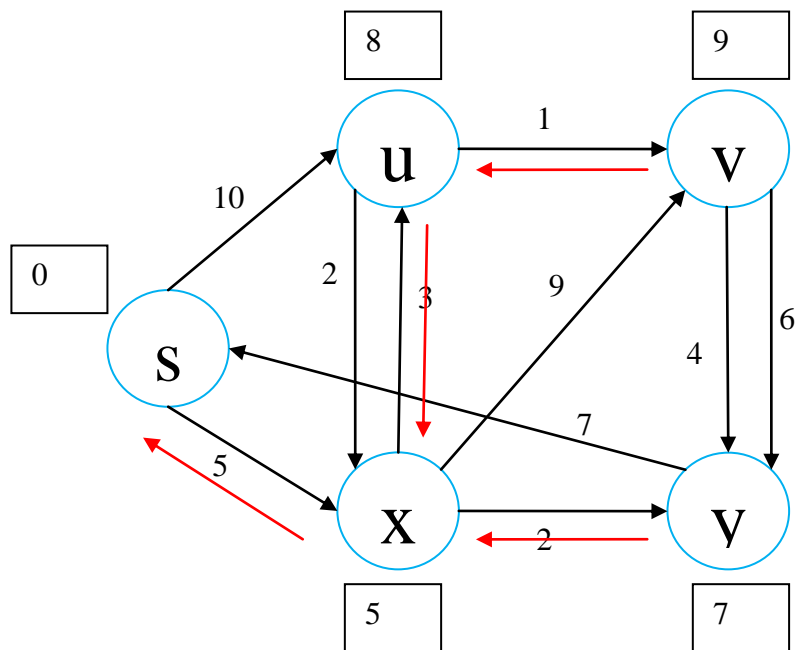
Slika 14: 4. korak algoritma

Sada je y najbliži, dodaje se u S . Otpusti se v i promijeni lista njegovih prethodnika.



Slika 15: 5. korak algoritma

Sada je u najbliži, odabere se i promijeni njegova lista prethodnika.



Slika 16: 6. korak algoritma

Konačno, dodaje se v . Lista prethodnika sada definira najkraći put od svakog čvora do s .

3.Facebook

Facebook[7] je društvena mreža sa slobodnim pristupom, osnovana u veljači 2004. godine. Osnovao ju je Mark Zuckerberg sa svojim kolegama Dustinom Moskowitzem i Chrisom Hughesom dok je bio student na Harvardu. U početku je bio namijenjen samo studentima na Harvardu, ali se poslije proširio na ostala sveučilišta, nakon toga na srednje škole, dok je danas pristup dozvoljen svima starijima od 13 godina. Trenutno ima više od 200 milijuna aktivnih korisnika (korisnici koji su posjetili stranicu u zadnjih 30 dana), od kojih 100 milijuna posjećuje stranicu jednom dnevno.

Facebook[8] je zamišljen tako da se putem njega povežu ljudi koji se već poznaju u stvarnom životu. Korisnici se mogu priključiti mrežama koje su organizirane po gradovima, radnim mjestima, školama i regijama. Također mogu dodavati prijatelje i slati im poruke, te mijenjati svoj osobni status.

Facebook je u svibnju 2007. objavio Facebook platformu[9], koja pruža *framework* za kreiranje aplikacija koje su mogu integrirati u Facebook. Trenutno postoji više od 52000 aplikacija objavljenih u kazalu aplikacija na Facebooku.

3.1.Facebook platforma

Facebook platforma se sastoji od više temeljnih komponenti, u koje se ubrajaju API, FBML, XFBML, FQL i FBJS. U nastavku su opisane najbitnije.

3.1.1.API

Pomoću Facebookovog API-ja moguće do dodati socijalni kontekst u aplikaciju koristeći profil, prijatelje, grupe, fotografije i podatke o događaju. Facebookov API koristi REST sučelje, što znači da su svi pozivi metoda ostvareni slanjem HTTP GET i POST zahtjeva.

3.1.2.FBML

FBML (Facebook Markup Language) omogućuje korisniku razvoj aplikacija na Facebook platformi koje se duboko integriraju u sam Facebook. Postoji više točaka integracije, od kojih su neke profil, *profile actions*, *canvas stranica*, *news feed* i *mini-meed*.

FBML je evoluirani podskup HTML-a. Neki elementi su uklonjeni, a neki koji su specifični za Facebook su dodani.

FBML ima više primjena u Facebooku, koristi se u sigurnim uvjetima na korisničkim profilima, u malim isječcima iz news feeda i za cijele *canvas* stranice. Cilj FBML-a je podržavanje raznovrsnog skupa tagova koji se mogu iskoristiti za različite ciljeve.

S izuzetkom nekoliko specifičnih, svi tagovi se mogu svrstati u sljedeće kategorije.

Tagovi sa socijalnim podacima

Predstavljaju srž FBML-a- Glavna uloga FBML-a je jednostavno umetanje socijalnih podataka u HTML. Socijalni podaci dohvaćaju i formatiraju podatke aplikaciji ili korisniku koji postavlja zahtjev. Ovi podaci mogu biti u više formi, kao što su informacije o korisniku, slike itd. Dobar primjer takvog taga je *fb:name*, koji identificira i vraća korisnikovo ime na više načina i link na njegov profil. Ključna specifičnost FBML-a je kontrola nad privatnim podacima. Sustav mora biti svjestan tko šalje zahtjev za podatkom i prema tome vraća odgovor. Npr. netko tko je logiran u Facebook tagom *fb:name* može dobiti puno ime i prezime, dok netko tko promatra van Facebooka dobiva samo ime, a ako je logirani korisnik blokiran, neće vidjeti ništa.

Sanitization tagovi

Sve web stranice imaju svoje standarde, čak unutar stranice različita područja mogu imati svoja pravila. Da bi se održao određeni izgled, postavke i privatnost stranice, FBML koristi ove tagove za pročišćavanje i uklanjanje određenih HTML tagova. Primjer je *fb:swf*, tag koji upravlja prikazom swf datoteka. Na korisničkom profilu se ne smiju pokretati automatski i prikažu se kao slika, dok se na *canvas* stranici se ponašaju kao normalni Flash objekt.

Dizajnerski tagovi

Dizajnerski tagovi pomažu definirati izgled stranice. FBML dozvoljava upotrebu HTML-a i CSS-a, ali postoje već definirani tagovi pomoću kojih je lakše uklopiti aplikaciju u stil Facebooka. Primjer je tag *fb:tabs*, koji stvara standardni Facebookov tab.

Tagovi komponenti

Tagovi komponenti pružaju bogatije mogućnosti na stranici, stvaraju komponente koje dopuštaju korisniku interakciju s aplikacijom na Facebooku. Ovi tagovi uglavnom daju gotovu komponentu koju je potrebno samo smjestiti u aplikaciju.

Primjer ovakvog taga je *fb:comments*, komponenta koja prikazuje ploču s komentarima u aplikaciji. Facebook se brine za prikaz komentara, izgled i sl. Nije potreban nikakav dodatni kod da bi komponenta radila.

Kontrolni tagovi

Postoje slučajevi kada se ne može unaprijed znati kako prikazati stvari u FBML-u i potrebno je ispitivati određene uvjete. FBML ima skup kontrolnih tagova za ovu primjenu. Primjer je tag *fb:visible-to-owner* koji dozvoljava da samo vlasnik profila vidi sadržaj u *profile boxu*. Kontrolni tagovi se ne koriste kao tagovi za osiguranje privatnosti, nego kao kontrola kako će FBML prikazati informacije na stranici. Sadržaj koji je skriven za korisnika nije zapravo uklonjen sa stranice, nego samo zaklonjen.

3.1.3.FQL

FQL (Facebook Query Language) je način za pristupanje socijalnim podacima s Facebooka pomoću sučelja koje je slično SQL-u. Do istih podataka se može doći i kroz metode API-ja.

Tipični upit izgleda ovako:

```
SELECT name, pic FROM user WHERE uid=211031 OR uid=4801660
```

Prednosti FQL-a pred API metodama su sljedeće:

- Sažeti XML smanjuje količinu prometa i troškove parsiranja.
- Kompleksniji zahtjevi mogu reducirati potreban broj zahtjeva.
- Dobiva se konzistentno, unificirano sučelje za sve podatke.

3.2.Točke integracije

Facebook aplikacija može doći u više oblika, međutim, preporuča se iskoristiti što više točaka integracije. Aplikacija se može integrirati u sljedeća područja na korisničkom profilu:

- **Kazalo aplikacija (Application Directory)**

U kazalu korisnika navedeni su osnovni podaci o aplikaciji. Da bi se aplikacija pojavila u kazalu, programer je mora poslati (*submit*) u kazalo zajedno s nekim osnovnim podacima (opis, logo...).

- **Stranica koja opisuje aplikaciju (About Page)**

Do stranice se dolazi iz kazala aplikacija. To je Facebook stranica koja opisuje aplikaciju.

- **Canvas stranica (Canvas Page)**

Canvas stranica je glavna stranica aplikacije na Facebooku. Pojavljuje se unutar Facebook okvira.

- **Kontrolna ploča (User Dashboard)**

Kontrolna ploča se obično nalazi na vrhu *canvas* stranice aplikacije. Pomoću nje korisnik može upravljati svojim sadržajem unutar aplikacije ili vidjeti informaciju o sadržaju drugog korisnika unutar aplikacije.

- **Profil**

- **Profile Box**

Profile Box je mjesto gdje se prikazuju najnoviji podaci o korisniku ili najnovije korisnikove akcije unutar aplikacije. Cilj *profile boxa* je da korisnik izrazi svoj identitet kroz aplikaciju.

- **Aplikacijski izbornik (Application Menu)**

Aplikacijski izbornik je mjesto gdje korisnici imaju pristup aplikaciji. Tu mogu mijenjati postavke aplikacije, otići na zadnje korištene aplikacije i sl.

- **Bookmarks**

Pojavljuju se na početnoj stranici korisnika i na kazalu aplikacije.

- **Tabovi aplikacija (Application Tab)**

Tabovi aplikacije su hibrid *canvas* stranice i *profile boxa*, na cijelom tabu otvara se *canvas* stranica aplikacije.

- **Tab Boxes (Boxes Tab)**

Na tabu Boxes aplikacija ima svoj *profile box*. Korisnici mogu profile boxove seliti s glavnog profila na tab *Boxes* i obratno.

- **Info sekcija aplikacije (Application Info Section)**

Info tab služi za strukturirano izražavanje korisnikovih interesa. Korisnici mogu dodati i podatke o aplikacijama u Info sekciju aplikacije. Info sekcija aplikacije sadrži tekst i/ili slike koje nudi aplikacija.

- **Publisher**

Publisher služi za objavljivanje informacija i poruku na korisnikovom feedu i na feedovima njegovih prijatelja. Aplikacije se mogu integrirati u *Publisher* tako da korisnici mogu stvarati i slati sadržaj.

- **Feed Forms**

Feed forms su FBML komponente koje služe aplikaciji za slanje feedova u korisnikovo ime. Aplikacija može slati direktno na korisnikov *mini-feed*, ili na *mini-feedove* njegovih prijatelja.

- **Postavke privatnosti (Privacy Settings)**

Korisnici ovdje mogu podesiti postavke privatnosti. Ove postavke uključuju i dozvole za objavu sadržaja.

- **Feed**

Aplikacija može pristupiti *feedu* korisnika i slati sadržaj na njega kao obavijest (*notification*).

- **Alerts**

Aplikacija može slati obavijesti (*notifications*) korisniku putem maila. Korisnici moraju dati aplikaciji dozvolu za to.

- **Zahtjevi (Requests)**

Aplikacije mogu stvarati zahtjeve koji se prikazuju na korisnikovoj početnoj stranici. Ovi zahtjevi su obično inicirani od korisnikovih prijatelja i zahtijevaju od korisnika da izvrši određenu akciju. Kao i sadržaj na *feedu* i zahtjevi se mogu slati kao obavijesti (*notifications*).

- **Interakcija s korisnicima (Interacting with Users)**

Korisnika se može usmjeriti na razne Facebook stranice da bi izveo određenu akciju.

3.3. Postavljanje Facebook aplikacije

Prvi korak u stvaranju Facebook aplikacije je logiranje u Facebook aplikaciju *Developer*, nakon što se aplikaciji da dopuštenje za pristup korisnikovom profilu, može se početi s postavljanje aplikacije. Nakon odabira opcije za stvaranje nove aplikacije dobiju se osnovni podaci za novu aplikaciju, koji uključuju:

- API ključ (*API key*) – kod koji identificira aplikaciju na Facebooku, šalje se skupa s svim pozivima API-ja
- Tajni kod (*secret*) – koristi ga Facebook za autentikaciju zahtjeva

Programer nakon toga mora popuniti sljedeća polja:

- *Canvas callback URL* – adresa aplikacije na serveru
- *Canvas page URL* – *canvas* stranica je adresa aplikacije na Facebooku. Kad korisnici pristupe aplikaciji na Facebooku, preusmjeri ih se na *canvas* stranicu aplikacije
- Mora li se aplikacija instalirati? – odabere se da ili ne.

Nakon ovog koraka, moguće je početi s razvojem aplikacije.

U sljedećoj tablici dan je pregled postavki aplikacije koje se mogu mijenjati u Developer aplikaciji. Nazivi su s hrvatskog i engleskog prijevoda Facebooka.

Tablica 4: Osnovne postavke Facebook aplikacije

Osnovno		
Naziv (engleski)	Naziv (hrvatski)	Opis
Application Name	Naziv aplikacije	
Description	Opis aplikacija	
Icon	Ikona	Mora biti veličine 16 × 16 piksela.
Logo	Logo	Mora biti veličine do 75 × 75 piksela.
Language	Jezik	
Developers	Developeri	Jedini mogu mijenjati aplikaciju i vidjeti je u <i>sandbox</i> načinu rada. Imena <i>developer</i> a se pojavljuju i na stranici o aplikaciji.
Developer Contact Email	Developer Contact Email	Email adresa na koju Facebook može kontaktirati developera
User Support Email	User Support Email	Email adresa na koju korisnici aplikacije mogu kontaktirati <i>developer</i> a.
Help URL	Help URL	Apsolutni URL na stranicu za pomoć pri korištenju aplikacije.
Privacy URL	Privacy URL	Apsolutni URL na stranicu s postavkama privatnosti za aplikaciju.
Terms of Service URL	Terms of Service URL	Apsolutni URL na stranicu za pravila o korištenju aplikacije.

Daljnje postavke nisu prevedene na hrvatski pa se navode samo engleski nazivi.

Tablica 5: Postavke autentičnosti Facebook aplikacije

Autentičnost	
Naziv (engleski)	Opis
Installable to? <ul style="list-style-type: none">• Users• Facebook Pages	Odabir vrste Facebook profila na kojeg se aplikacija može instalirati.
Post-Authorize Callback URL	URL kojeg se <i>pinga</i> nakon što korisnik prvi put autorizira aplikaciju.
Post-Remove Callback URL	URL kojeg se <i>pinga</i> nakon što korisnik ukloni aplikaciju.

Tablica 6: Postavke profila

Profili	
Naziv (engleski)	Opis
Profile Tab	
Tab Name	Labela za aplikacijski tab na profilu.
Tab URL	URL s kojeg Facebook dohvaća sadržaj za aplikacijski tab
Profile Box	
Default Column <ul style="list-style-type: none"> • Wide • Narrow 	Definiranje mjesta gdje se <i>Profile Box</i> inicijalno pojavljuje.
Info Section	
Information Update Callback URL	URL kojeg Facebook <i>pinga</i> nakon što korisnik promijeni podatke na info sekciji aplikacije na profilu
Profile Publisher	
Publish Text	Labela za aplikacijski link na aplikaciju <i>Publisher</i> za kreiranje sadržaja na profilima prijatelja
Publish Callback URL	URL s kojeg Facebook dohvaća sadržaj za objavljivanje na <i>Publisheru</i> na profilima korisnikovih prijatelja
Self-Publish Text	Labela za aplikacijski link na aplikaciju <i>Publisher</i> za kreiranje sadržaja na korisnikovom profilu
Self-Publish Callback URL	URL s kojeg Facebook dohvaća sadržaj za objavljivanje na <i>Publisheru</i> na korisnikovom profilu
Publisher Mode <ul style="list-style-type: none"> • Inline • Dialog(Legacy) 	Odabir širine piksela za prikaz sadržaja na <i>Publisheru</i> , ako je širina između 370px i 430px odabire se <i>Inline</i> , inače <i>Dialog</i> .
Facebook Page Administration	
Edit URL	URL na kojem administratori aplikacije mogu promijeniti postavke aplikacije

Tablica 7: Postavke *canvas* stranice

Canvas	
Naziv (engleski)	Opis
Optional URLs	
Bookmark URL	URL koji se otvara nakon što korisnik otvori aplikaciju preko prečice. Ako se ne unese, to je <i>canvas</i> stranica.
Post-Authorize Redirect URL	URL na kojeg korisnik prelazi nakon što je prvi put autorizirao aplikaciju. Ako se ne unese, to je <i>canvas</i> stranica.
Canvas Settings	
Render Method <ul style="list-style-type: none"> • FBML • IFrame 	Odabir prikaza <i>canvas</i> stranice.
IFrame Size <ul style="list-style-type: none"> • Smart Size • Resizable 	Za prikaz pomoću <i>IFramea</i> odabire se način postavljanja veličine <i>IFramea</i> .
Canvas Width <ul style="list-style-type: none"> • Full width (760px) • Old width (646px) 	Odabir veličine <i>canvas</i> stranice, ili puna veličina od 760px ili po starom standardu 646px.
Quick Transitions <ul style="list-style-type: none"> • On • Off 	Omogućavanje brzog prijelaza na <i>canvas</i> stranicama pomoću preskakanja otvaranja Facebook okvira.

Tablica 8: Dodatne postavke

Dodatno	
Naziv (engleski)	Opis
Advanced Settings	
Application Type <ul style="list-style-type: none"> • Web • Desktop 	Odabir vrste aplikacije, desktop ili web aplikacija.
Sandbox Mode <ul style="list-style-type: none"> • Enable • Disable 	Omogućavanje vrste rada pri kojoj samo developeri mogu pristupiti aplikaciji.
Mobile Integration	
SMS Integration <ul style="list-style-type: none"> • Enable • Disable 	Omogućavanje integracije aplikacije sa SMS porukama.
Attachments	
Attachment Text	Labela za URL za dodavanje sadržaja uz aplikaciju.
Attachment Callback URL	URL s kojeg aplikacija dohvaća sadržaj za sučelje za dodavanje sadržaja uz aplikaciju.
Security	
Server Whitelist	Facebook blokira zahtjeve koji dolaze s IP adresa koje nisu u ovoj listi.
JavaScript API Library Whitelist Exception <ul style="list-style-type: none"> • Enable • Disable 	Omogućavanje <i>IFrame</i> aplikacijama da zaobiđu <i>Server Whitelist</i> ako se koristi.
Legal	
Video Rentals Does your site publish feed stories involving video sales, rentals, or subscriptions? <ul style="list-style-type: none"> • Yes • No 	Ako se odabere „Yes“, stranica mora poštovati Video Privacy Protection Act

4.Implementacija

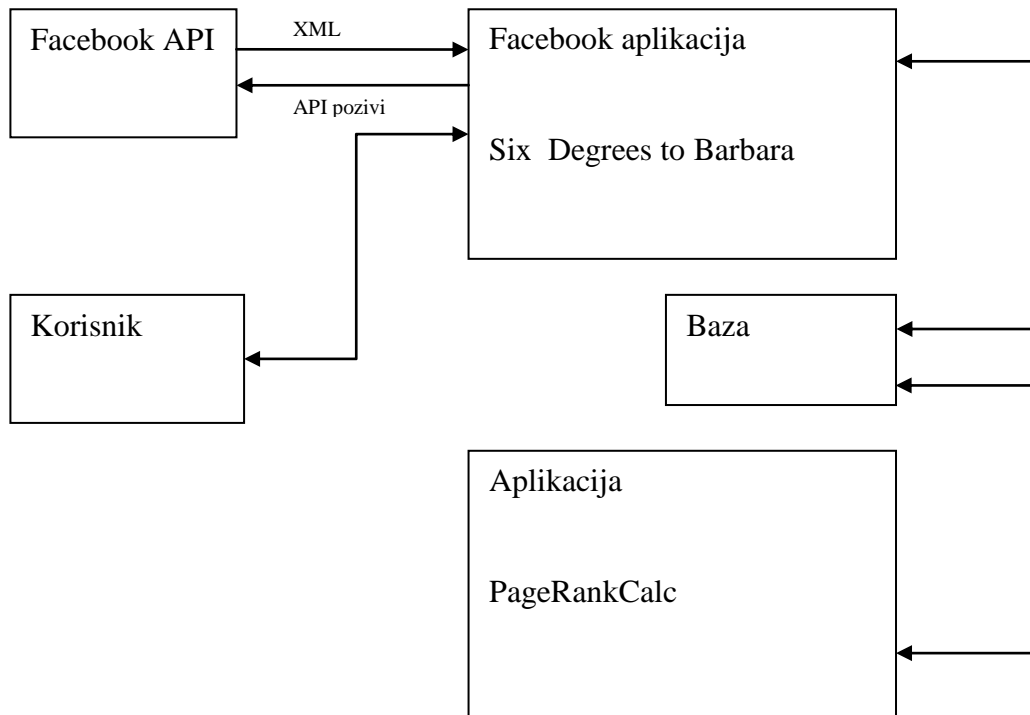
Razvijene su dvije odvojene aplikacije, zbog odvajanja prikaza rezultata od računanja, te radi lakšeg kasnijeg unapređivanja. Prva je Facebook aplikacija Six Degrees to Barbara, a druga PageRankCalc.

Aplikacija Six Degrees to Barbara je web aplikacija, ostvarena uz pomoć Pythonovog[10] web *frameworka* Django[11], uz korištenje biblioteke PyFacebook[12] i MySQLdb[13]. PyFacebook je biblioteka za interakciju s Facebook API-jem, a MySQLdb biblioteka za interakciju s MySQL[14] bazom podataka. Aplikacija je smještena je na zavodskom serveru. Posluhuje se s Apache[15] web servera.

Aplikacija PageRankCalc je razvijena u programskom jeziku Python, te radi i pod Linux i pod MS Windows operacijskim sustavima. Korištene su biblioteke igraph[16] (za prikaz grafa i računanje PageRanka), te MySQLdb za komunikaciju s bazom.

Sav softver korišten pri izradi programa je besplatan i otvorenog izvornog kôda.

4.1. Arhitektura



Slika 17: Arhitektura

Na slici gore prikazana je arhitektura aplikacije. Sastoji se od dva neovisna dijela, Facebook aplikacije Six Degrees to Barbara koja komunicira s Facebook API-jem i bazom i aplikacije PageRankCalc koja računa PageRank vrijednosti.

- **Korisnik**

Korisnik pristupa Facebook aplikaciji tako da otvori link na nju prilikom prvog posjeta, a kasnije ima poveznicu na svojim stranicama. Također na izborniku *Boxes* na korisnikovom profilu nalazi se prozor aplikacije u kojem se ispisuje korisnikov rang izračunat po PageRank algoritmu i Barbara number.

- **Facebook aplikacija Six Degrees to Barbara**

Prilikom dodavanja novog korisnika aplikacija zapisuje njegove podatke u bazu. Odmah se računa Barbara number i ispisuje u aplikaciji i u *Boxu*, dok se korisnikov rang ispisuje nakon sljedećeg redovnog računanja. Lista korisnikovih prijatelja se ažurira u bazi nakon zadanog vremenskog razdoblja.

- **Facebook API**

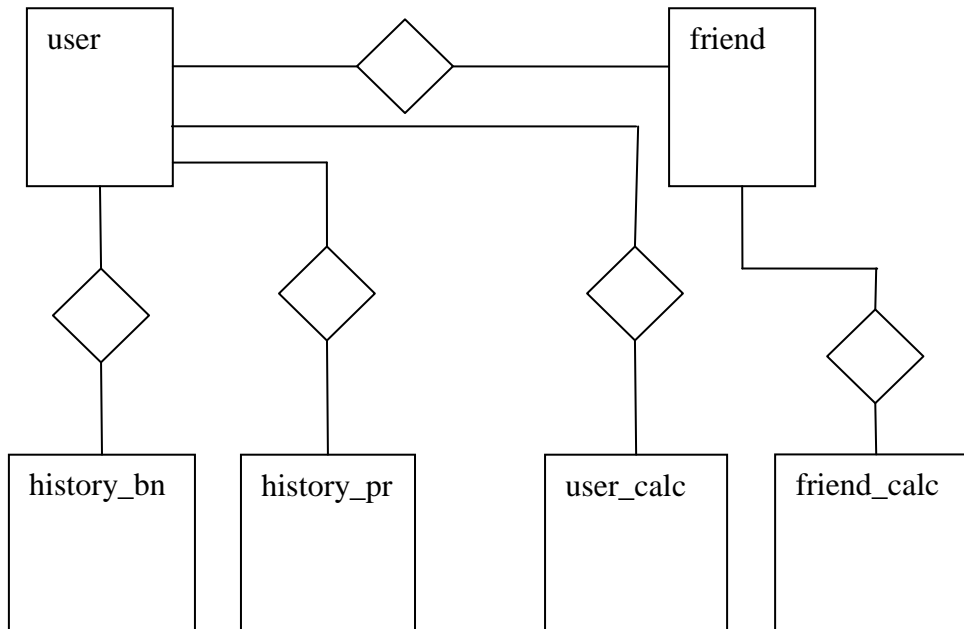
Aplikacija poziva metode Facebook API-ja, koji šalje odgovore u XML formatu

- **Aplikacija PageRankCalc**

Aplikacija periodički računa PageRank i Barbara number i zapisuje rezultate u bazu.

4.2. Baza

Koristio se MySQL[] sustav za upravljanje bazom podataka. Na slici ispod prikazan je ER model baze.



Slika 18: ER model baze

U tablici *user* sadržani su podaci o korisnicima aplikacija.

Tablica 9: Opis tablice user

Naziv stupca	Tip podatka	Opis
id	INT(10)	korisnikov Facebook id, primarni ključ
date_added	DATETIME	datum kad je korisnik dodao aplikaciju
pagerank	FLOAT	izračunati korisnikov PageRank
barbara_number	INT(10)	duljina najkraćeg puta do Barbare
date_pr	DATETIME	datum posljednjeg izračuna PageRanka
date_bn	DATETIME	datum posljednjeg izračuna Barbara numbera
date_friends	DATETIME	datum posljednjeg dohvaćanja liste korisnikovih prijatelja
rank	INT(10)	korisnikov rang izračunat po PageRank algoritmu

Tablica *friend* sadrži podatke o korisnikovim prijateljima.

Tablica 10: Opis tablice friend

Naziv stupca	Tip podatka	Opis
id	INT(10)	korisnikov Facebook id, dio kompozitnog primarnog ključa
friend_id	INT(10)	Facebook id korisnikovog prijatelja, dio kompozitnog primarnog ključa

U tablicama *history_pr* i *history_bn* vodi se evidencija o prošlim vrijednostima PageRanka i Barbara numbera. Ove tablice se pune pomoću okidača na tablici user. Pri svakom mijenjanju PageRanka i Barbara numbera, stare vrijednosti i datum promjene se upisuju u *history_pr* i *history_bn*.

Tablica 11: Opis tablice history_bn

Naziv stupca	Tip podatka	Opis
id	INT(10)	korisnikov Facebook id, dio kompozitnog primarnog ključa
date_bn	DATETIME	datum izračuna novog Barbara numbera, dio kompozitnog primarnog ključa
barbara_number	INT(10)	duljina najkraćeg puta do Barbare

Tablica 12: Opis tablice history_pr

Naziv stupca	Tip podatka	Opis
id	INT(10)	korisnikov Facebook id, dio kompozitnog primarnog ključa
date_pr	DATETIME	datum izračuna novog PageRanka, dio kompozitnog primarnog ključa
PageRank	FLOAT	korisnikov PageRank
rank	INT(10)	korisnikov rang izračunat po PageRank algoritmu

Tablice *user_calc* i *friend_calc* služe kao pomoć aplikaciji PageRankCalc za računanje PageRank vrijednosti. Prije računanja PageRanka potrebno je osvježiti podatke u njima. U tu svrhu stvorene su procedure *precalc* i *precalc2* koje se pozivaju iz aplikacije PageRankCalc.

Tablica 13: Opis tablice *user_calc*

Naziv stupca	Tip podatka	Opis
id	INT(10)	korisnikov Facebook id, primarni ključ
id_calc	INT(10)	id korisnika koji se koristi u računanju PageRanka

Tablica 14: Opis tablice *friend_calc*

Naziv stupca	Tip podatka	Opis
id_calc	INT(10)	id korisnika koji se koristi u računanju PageRanka, dio kompozitnog primarnog ključa
friend_id_calc	INT(10)	id korisnikovih prijatelja, koristi se u računanju PageRanka, dio kompozitnog primarnog ključa

4.3. Izgled i sučelja aplikacija

4.3.1. Grafičko sučelje aplikacije Six Degrees to Barbara

Aplikacija se integrira u Facebook pomoću *canvas* stranice, stranice o aplikaciji, *boxa* na profilu i okvira na tabu *Boxes*.

Canvas stranica se sastoji od dva taba. Prvi tab je početna stranica aplikacije, a u drugom je prikazana korisnikova statistika za zadnjih 6 mjeseci. Na slici ispod je prikazana početna stranica aplikacije. Na njoj je ispisana i vizualno predložena korisnikov Barbara number, te njegov rang. S desne strane u okviru nalazi se poveznica kojom korisnik dodaje aplikaciju u *box* na profilu ili u okvir na tabu *Boxes*. Ispod nje nalaze se imena i slike korisnikovih

prijatelja koji su već dodali aplikaciju, a ispod njih poveznica na stranicu na kojoj se može prijateljima poslati poziv za korištenje aplikacije.



Slika 19: Početna stranica aplikacije

Na drugom tabu nalazi se korisnikova statistika. U prvoj tablici ispisana je statistika za rang, a u drugoj za Barbara number. Okvir desno ima iste funkcionalnosti kao na početnoj stranici.

facebook Home Profile Friends Inbox Pero Zdero Settings Logout

Home History



Hello Pero, this is your popularity history.

Date	Rank
2009-05-03 23:46:00	4
2009-05-03 23:45:52	4
2009-05-03 23:44:54	4
2009-05-03 23:44:39	4
2009-05-03 23:38:42	5
2009-05-03 23:37:18	4
2009-05-03 23:36:58	5
2009-05-03 23:36:34	4
2009-05-03 23:32:27	2
2009-05-03 23:30:00	4
2009-05-03 22:50:59	4
2009-05-03 22:50:45	4
2009-05-03 22:49:32	4
2009-05-03 22:37:22	2
2009-05-03 14:42:29	2
2009-05-03 14:42:26	2

This is the history of your distance from Barbara.

Date	Distance
2009-05-03 23:46:00	2
2009-05-03 23:45:27	2
2009-05-03 23:44:54	2
2009-05-03 23:44:39	2
2009-05-03 23:38:42	2
2009-05-03 23:37:18	2

ADD TO PROFILE

YOUR FRIENDS:



Barbara Carević

INVITE MORE FRIENDS



Applications

Slika 20: Tab sa statistikama

Na stranici za pozivanje prijatelja nalazi se izbornik pomoću kojeg se odaberu prijatelji kojima se želi poslati pozivnica.




You can invite your friends to use Six Degrees To Barbara.

Start Typing a Friend's Name

- Aba Foxic Lady
- Ana Beblek
- Ana Charlotte Crimson
- Ana Gudalović
- Ana Soce
- Ana Zivic


You haven't selected anyone.

 Send Invitation

Slika 21: Stranica za slanje pozivnica

Na stranici o aplikaciji dane su osnovne informacije o aplikaciji, navedeni su programeri koji su je razvili, te korisnici koji su je dodali.

6 Six Degrees to Barbara
◀ Browse More Applications



Welcome to Six Degrees to Barbara!


This application calculates your popularity on Facebook using Google's PageRank. You can find out more about the algorithm on <http://en.wikipedia.org/wiki/PageRank>. It also shows your distance from the application's developer.

Spread the app to become more popular!


Facebook is providing links to these applications as a courtesy, and makes no representations regarding the applications or any information related to them. Any questions regarding an application should be directed to the developer.

▼ Friends Who Have Added this Application


3 friends have added this application See All



Pero Zdero



Barbara SixDegrees



Iva Crimson

▼ Discussion Board

There are no discussions.
Start the first topic.

▼ Reviews

Displaying reviews Write Review | See All

▼ The Wall

No wall posts See All

Write something...


▼ About this Application

Users:
0 monthly active users,
3 friends

Category
Education

This application was **not** developed by Facebook.

▼ About the Developer



Barbara SixDegrees
(Croatia)

Contact Developer | Report Application

Slika 22: Stranica o aplikaciji

Box u profilu i okvir u tabu *Boxes* izgledaju jednako. Naveden je korisnikov Barbara number i rang.



Slika 23: Box na korisnikovom profilu

Za obradu slika korišten je GIMP[17], a grafički elementi preuzeti su sa stranice MediaSlap[18].

4.3.2. Sučelje aplikacije PageRankCalc

Aplikacija PageRankCalc se pokreće iz naredbene linije, sa ili bez parametara. Postoje četiri načina pozivanja.

- *python PageRankCalc*

Aplikacija će izračunati PageRank i Barbara number.

- *python PageRankCalc -bn*

Aplikacija će izračunati samo Barbara number.

- *python PageRankCalc -pr*

Aplikacija će izračunati samo PageRank.

- *python PageRankCalc -bn -pr* ili *python PageRankCalc -pr -bn*

Aplikacija će izračunati PageRank i Barbara number.

Nakon svakog izvođenja aplikacije podaci o izvođenju se zapisuju u datoteku PageRankCalc.log.

Podaci koji se zapisuju su vrijeme početka računanja, izvršeni izračuni, vrijeme izvođenja, te maksimalni Barbara number (ako se računa).

Primjer zapisa u PageRankCalc.log datoteci:

Started at: 2009-05-04 21:25:10.843000

PageRank calculated in 0.0310001373291

Maximum distance to vertex 0: 3

Barbara number calculated in 0.0160000324249

4. Diskusija

Najveći problem prilikom izrade programa bila je veličina grafa. Bilo je potrebno naći gotovu implementaciju PageRanka koja radi s jako velikim grafovima, budući da se u budućnosti očekuje veliki broj korisnika. U prosjeku svaki korisnik Facebooka ima 120 prijatelja, što je dosta više od 10 izlaznih veza po stranici, koliko ima prosječna web stranica. Također treba uzeti u obzir da su sve veze na Facebooku dvosmjerne, tj. ako korisnik A doda korisnika B kao prijatelja, onda i korisnik B automatski ima kao prijatelja korisnika A. Problem je u zauzeću radne memorije, budući da je algoritam takav da cijeli graf mora biti dostupan. Najboljom se pokazala implementacija iz biblioteke *igraph*, koja na računalu s 4 GB RAM-a radi do grafova s otprilike 1500000 vrhova s testnim podacima. Ova implementacija koristi vezane liste. To se riješilo uvođenjem ograničenja. Ako u jednom trenutku graf bude imao više od 1500000 vrhova (u taj broj ulaze korisnici koji su dodali aplikaciju i njihovi prijatelji koji nisu). Prvo se iz grafa izbacuju svi koji nisu korisnici, ako je preostali broj vrhova i dalje veći od 1500000, računa se maksimalni Barbara number za koji je korisnika s tim Barbara numberom ili manjim manje od 1500000, te se samo od takvih korisnika kreira graf.

Za efikasno računanje PageRanka[19] za velike grafove, postoje tri pristupa.

- Komprimiranje ulaznih podataka uz prilagodbu algoritma

Najčešće prilagodbe se odnose na dijeljenje grafa na povezane podgrafova. Ova prilagodba u ovom slučaju nema smisla, budući da se očekuje da će skoro cijeli graf biti dobro povezan. Druga česta prilagodba je drugačije rukovanje stranicama koje nemaju izlaznih veza[20], što također nema smisla u ovom slučaju jer su sve veze dvosmjerne.

- Računanje s dijelovima grafa koje zahtijeva pohranjivanje međurezultata na disk

Ova metoda rezultira velikim brojem čitanja i pisanja što značajno produljuje vrijeme izvođenja. Da bi se upotrijebila bilo bi potrebno prilagoditi prikaz grafa u memoriji i bazi.

- Paralelno (distribuirano) računanje[21]

Svodi se na metodu s pohranjivanjem međurezultata na disk, koja se ubrzava paralelnim izvođenjem.

Trenutno ne postoje gotove implementacije koje bi radile na neki od gore navedenih načina.

Dodatne probleme je predstavljalo što unaprijed nije bilo moguće predvidjeti međusobnu povezanost korisnika koji će koristiti aplikaciju, pa je teško procijeniti kvalitetu testnih podataka.

6.Zaključak

U sklopu ovog rada razvijena je Facebook aplikacija Six Degrees to Barbara koja prikazuje korisnikovu popularnost, tj. njegov rang izračunat po PageRank algoritmu u grafu koji se sastoji od korisnika aplikacije i njihovih prijatelja. Aplikacija također prikazuje udaljenost korisnika od autora aplikacije (Barbara number). Prikazana je i statistika korisnikovog ranga i Barbara numbera za zadnjih 6 mjeseci. Zbog preglednosti i mogućih unapređenja u budućnosti aplikacija koja računa PageRank i Barbara number PageRankCalc realizirana je odvojeno od Facebook aplikacije. PageRankCalc se neće pokretati svaki put kad novi korisnik doda aplikaciju, nego periodički radi memorijskih zahtjeva. Aplikacija Six Degrees to Barbara, koja je smještena na zavodskom serveru, dohvaća podatke o korisnicima koristeći Facebookov API i pohranjuje ih u bazu. Aplikacija PageRankCalc ih dohvaća iz baze, te koristi za svoje izračune koje također sprema u bazu.

Facebook nema ugrađeno mjerenje popularnosti, pa bi aplikacija mogla biti zanimljiva korisnicima. Svaki korisnik može poslati pozivnice za dodavanje aplikacije svojim prijateljima što olakšava širenje aplikacije.

Pokazalo se da za posluživanje velikog broja korisnika postojeće gotove implementacije PageRank algoritma neće biti dovoljne, te bi u budućnosti trebalo razviti bolju implementaciju za ovu primjenu. Trenutna implementacija radi za graf od 1500000 ljudi, a u taj broj ulaze korisnici aplikacije i njihovi prijatelji. Ako se dogodi da graf postane veći, prvo se izbacuju oni koji nisu dodali aplikaciju. Ako je graf i dalje prevelik, računa se maksimalni Barbara number za koji je korisnika s tim Barbara numberom ili manjim manje od 1500000, te se samo od takvih korisnika kreira graf. U budućnosti, ako broj korisnika naraste, trebalo bi razviti novu implementaciju PageRanka koja bi odgovarala ovim potrebama.

7.Literatura

- [1] Sergey Brin and Lawrence Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine, 1998, <http://infolab.stanford.edu/~backrub/google.html>
- [2] Wikipedia: PageRank, <http://en.wikipedia.org/wiki/PageRank>
- [3] David Austin: Google Page Rank Explained, <http://rahulhackingarticles.wetpaint.com/page/Google+Page+Rank+Explained>
- [4] ICS 161: Design and Analysis of Algorithms: Shortest Paths, 1996, <http://www.ics.uci.edu/~eppstein/161/960208>.
- [5] Dijkstra's Algorithm: <http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/GraphAlgor/dijkstraAlgor.htm>
- [7] Facebook: Press, <http://www.facebook.com/press/info.php?statistics>
- [8] Wikipedia: Facebook, <http://en.wikipedia.org/wiki/Facebook>
- [9] Facebook Developers, <http://developers.facebook.com/>
- [10] Python, 1990-2009, <http://www.python.org/>
- [11] Django, <http://www.djangoproject.com/>
- [12] PyFacebook, <http://code.google.com/p/pyfacebook/>
- [13] MySQLdb, <http://sourceforge.net/projects/mysql-python>
- [14] MySQL, <http://www.mysql.com/>
- [15] Apache, <http://www.apache.org/>
- [16] Igraph, <http://igraph.sourceforge.net/>
- [17] Gimp, <http://www.gimp.org/>
- [18] MediaSlap, <http://mediaslap.com/>
- [19] Amy N. Langville, Carl D. Meyer: Google's PageRank and Beyond: The Science of Search Engine Rankings, 2006
- [20] Adhemar Bultheel: Google's Eigenvector: The secret of PageRank, 2007, <http://www.kvab.be/Downloads/Lezingen/bultheel.pdf>
- [21] Christian Kohlschutter, Paul-Alexandru Chirita, and Wolfgang Nejdl: Efficient Parallel Computation of PageRank, <http://www.l3s.de/~chirita/publications/kohlschuetter06efficient.pdf>