Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Ervin Perhaj

Vizualizacija algoritma PageRank

diplomsko delo
visokošolskega strokovnega študija prve stopnje

Ljubljana, 2013

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Ervin Perhaj

Vizualizacija algoritma PageRank

diplomsko delo
visokošolskega strokovnega študija prve stopnje

Mentor: izr. prof. dr. Marko Robnik Šikonja

Ljubljana, 2013

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00384/2013
Datum: 02.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ERVIN PERHAJ**

Naslov: **VIZUALIZACIJA ALGORITMA PAGERANK**
VISUALIZATION OF PAGERANK ALGORITHM

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Pri poučevanju algoritmov za določanje pomembnosti vozlišč v usmerjenih grafih potrebujemo orodja za boljšo predstavitev vsebin in za njihovo lažje razumevanje. Rangiranje spletnih strani je inačica tega problema, za katero je bilo predlaganih več pristopov. Algoritem PageRank iterativno utežuje vozlišča glede na pomembnost strani, ki kažejo na dano stran. Njegovo delovanje je lažje razumeti z vizualizacijo na izbranih primerih.
Proučite delovanje algoritma PageRank in njegovo delovanje ilustrirajte na nekaj primerih grafov. Izdelajte animirano interaktivno spletno aplikacijo, ki bo uporabnika poučila o delovanju algoritma.

Mentor:
izr. prof. dr. Marko Robnik Šikonja

Dekan:
prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Izjava o avtorstvu diplomskega dela

Ervin Perhaj, z vpisno številko **63080374**, sem avtor diplomskega dela z naslovom:

Vizualizacija algoritma PageRank.

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Marka Robnika Šikonje,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela in
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 15. oktober 2013 Podpis avtorja:

Zahvaljujem se mentorju izr. prof. dr. Marku Robniku Šikonji, za pomoč in svetovanje pri izdelavi diplomskega dela.

Hvaležen sem mojim staršem, prijateljem in vsem, ki so me podpirali in spodbujali pri študiju.

Kazalo

Povzetek

Cilj diplomskega dela je razviti spletno aplikacijo, ki bo uporabniku pomagala razumeti delovanje algoritma PageRank.

Diplomsko delo smo razdelili na dva dela. Najprej smo razvili algoritem za računanje PageRank vrednosti spletnih strani. Algoritem na vhodu prejme seznam spletnih strani ter njihovih povezav, ki jih uporabnik vnaša prek spletnega vmesnika. Na podlagi teh podatkov izračuna vrednost PageRank za posamezno stran. Algoritem ponavlja postopek, dokler razlika PageRank vrednosti trenutne iteracije ter predhodne iteracije ni manjša od $0,0001$.

V drugem delu smo razvili vizualizacijo algoritma PageRank in spletni vmesnik, preko katerega uporabnik zgradi svoje omrežje ali izbere eno od že zgrajenih. Spletne strani so prikazane kot usmerjeni oz. neusmerjeni grafi, velikost vozlišča pa predstavlja vrednost PageRank.

Ključne besede

Algoritem PageRank, rangiranje spletnih strani, spletna aplikacija, vizualizacija

Abstract

The goal of the thesis is to develop a web application that help users understand the functioning of the PageRank algorithm.

The thesis consists of two parts. First we develop an algorithm to calculate PageRank values of web pages. The input of algorithm is a list of web pages and links between them. The user enters the list through the web interface. From the data the algorithm calculates PageRank value for each page. The algorithm repeats the process, until the difference of PageRank values between iterations is less than $0,0001$.

In the second part, we develop visualization of PageRank algorithm. The user can build his/her own network or select one of precreated ones. Web pages are represented as directed or undirected graphs, the size of the nodes represents PageRank value.

Keywords

PageRank algorithm, website ranking, web application, visualization

Poglavje 1

Uvod

Svetovni splet je v današnjem času nepogrešljiv. Z njim se srečujemo tako na delovnem mestu kot v prostem času. Uporabniki bi radi v čim krajšem času dobili dobre in zanesljive podatke. Iskalnik nam vrne veliko spletnih strani oziroma zadetkov, zato je pomembno, kako rangirati zadetke. Iskalniki se financirajo tudi s pomočjo oglaševanja, zato je zelo pomembna izbira dobrega rangirnega algoritma, saj bo zanesljive iskalnike uporabljalo veliko uporabnikov in so tako zanimivejši za oglaševalce[2]. Najpopularnejši iskalnik je Google. PageRank je bil Googlov sistem določanja "popularnosti" spletnih strani. Sedaj Google uporablja predvsem algoritma Panda in Penguin, s katerima kaznuje strani ali odstrani iz rezultatov iskanj, če se izkaže, da so si visoko pozicijo pridobile preko goljufije (angl. black hat SEO).

Delovanje algoritmov je mnogokrat težko razumeti, zato so dobrodošle njihove vizualizacije. V diplomskem delu predstavimo algoritem PageRank in razvijemo njegovo vizualizacijo. Za razvoj spletne aplikacije smo izbrali programski jezik Java Script. Za aplikacijo smo uporabili še označevalni jezik HTML5 v kombinaciji s slogovnim jezikom CSS, s katerima je izdelano spletno okolje za vizualizacijo algoritma.

V nadaljevanju predstavimo implementacijo vizualizacije algoritma PageRank in gradnje spletne aplikacije. V 2. poglavju opišemo algoritem PageRank in tudi druge načine, ki se danes uporabljajo za rangiranje in op-

timizacijo spletnih strani. V 3. poglavju predstavimo orodja in tehnologije za implementacijo spletne aplikacije in predstavimo njen razvoj. Četrto poglavje prikaže uporabo in delovanje spletne aplikacije. V zadnjem poglavju povzamemo narejeno in predstavimo nekaj idej za izboljšave.

Poglavje 2

Rangiranje spletnih strani

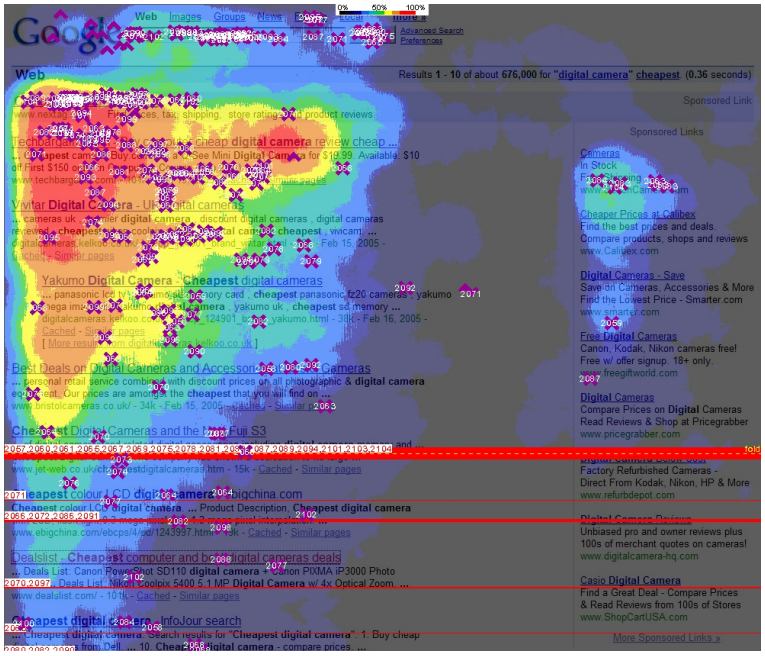
V tem poglavju predstavimo različne pristope za rangiranje spletnih strani. Glavni del poglavja predstavlja opis algoritma PageRank.

2.1 Optimizacija spletnih strani

Optimizacija (SEO – search engine optimization) (Slika 2.2) je postopek, s katerim lastniki želijo izboljšati pozicijo spletne strani v iskalnikih. Spletna stran se zato pojavi višje v rezultatih iskanj in posledično jo obišče več uporabnikov. Rezultati raziskav kažejo, da se pogledi uporabnikov zadržujejo v t.i. zlatem trikotniku (Slika 2.1). To pomeni, da se uporabnikov pogled začne v zgornjem levem kotu zadetkov, nato pa preleti prva dva ali tri zadetke iskanja. Vsak iskalnik ima svoja pravila razvrščanja zadetkov. Nekatera pravila so javno dostopna, druga pa so strogo varovana skrivnost. Pravila se stalno spreminjajo, zato je treba stran redno prilagajati in posodabljati. Najlažje je optimizacijo izvajati pri gradnji spletne strani od začetka, težje pa jo je izvajati na že postavljeni strani, ker to ne pomeni samo spremembe vsebine, ampak tudi oblike in videza spletne strani. Pri optimiziranju spletne strani postane stran lepše oblikovana, bolj pregledna in lažje berljiva [3, 12].

Obstajajo številne tehnike za dvig spletne strani v rezultatih iskalnikov. Delimo jih glede na kraj optimizacije:

- na spletni strani sami (on-site optimizacija),
- izven spletne strani (off-site optimizacija).



Slika 2.1: "Zlati trikotnik", kamor zahaja največ pogledov obiskovalcev.

2.1.1 Optimizacija na spletnem mestu

Optimizacija na spletni strani je prvi korak pri optimizaciji in uvrščanju spletne strani na višje mesto pri rezultatih iskanja. Optimizacijo spletne strani lahko podjetje oziroma oblikovalec naredi samostojno, lahko pa uporabi določene programe. Pomembno je, da se optimizira vsaka stran posebej in ne samo glavna vhodna stran. Takšna optimizacija je prilagoditev elementov na spletni strani:

- definiranje ključnih besed,
- obogatitev besedila s ključnimi besedami,

- ustrezno poimenovanje menijev,
- ustrezno poimenovanje slik,
- ustrezni URL naslovi,
- ustrezne meta povezave,
- notranje povezave.

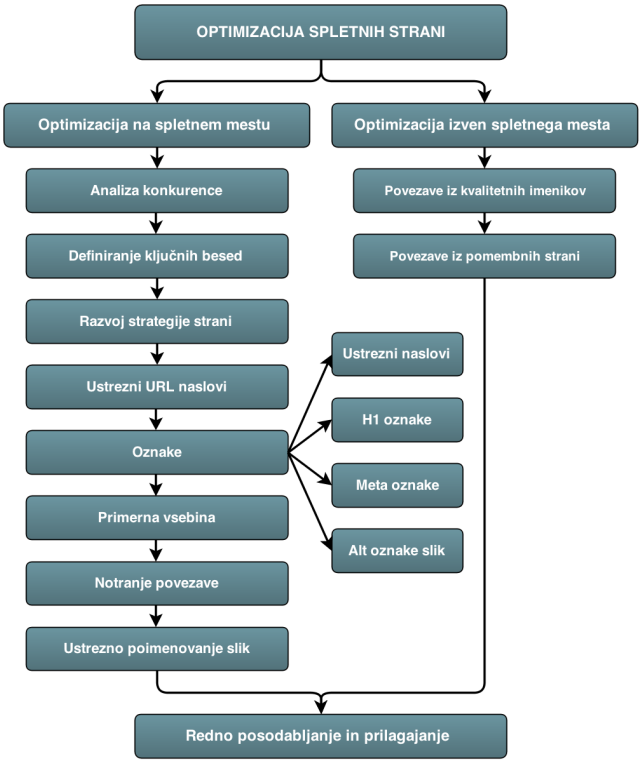
2.1.2 Optimizacija izven spletnega mesta

Optimizacija izven spletnega mesta se bolj ukvarja z zunanjimi dejavniki, ki vplivajo na uvrstitev spletne strani v rezultatih. Predvsem gre za povezave, ki kažejo na spletno stran iz drugih, kakovostnejših spletnih mest. Dohodne povezave naj bi prihajale s strani, s sorodnimi ključnimi besedami in tematiko. V to vrsto optimizacije spada tudi algoritem PageRank, ki mu bomo več pozornosti posvetili v naslednjem podpoglavju. Zunanji dejavniki so:

- število zunanjih povezav,
- kvaliteta zunanjih povezav,
- hitrost pridobivanja zunanjih povezav,
- starost povezave in domene,
- popularnost domene,
- besedilo na zunanji povezavi.

Zunanje povezave morajo biti pridobljene po pravični, naravni poti (angl. white hat SEO) in ne umetno oz. prek goljufije (angl. black hat SEO). SEO je angleška kratica za "*search engine optimization*" in pomeni optimizacija spletnih strani. Gre za pridobivanje dohodnih povezav s pomočjo farm povezav (angl. link farms). Te spletne strani vsebujejo izhodne povezave in tako dvigujejo oceno stranem, na katere kažejo. Poudariti je treba, da ima

metoda "white hat SEO" boljše in dolgoročne razultate, za razliko od metode "black hat SEO", ki deluje večinoma kratkoročno, ker algoritmi za rangiranje spletnih strani prepoznajo tak način pridobivanja povezav in te strani potem kaznujejo [12].



Slika 2.2: Prikaz optimizacije spletne strani.

2.2 Algoritem PageRank

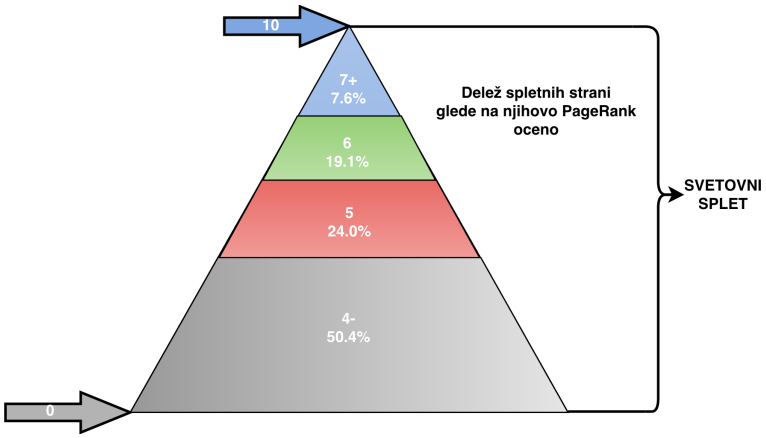
Vsak iskalnik ima v ozadju algoritem, s katerim ocenjuje spletne strani. Ponavadi gre za več algoritmov, ki delujejo na vseh nivojih iskanja, od indeksiranja spletnih strani do prikaza rezultatov. Lastniki iskalnikov te algoritme skrivajo, saj so konkurenčna prednost iskalnika. Algoritme podjetja spreminjajo in prilagajajo, da se njihovo delovanje izboljša.

2.2.1 Začetki algoritma PageRank

PageRank je bil razvit na univerzi Stanford v Kaliforniji, kot del raziskovalnega projekta "Internetni iskalniki nove generacije". Razvila sta ga Larry Page in Sergey Brin leta 1996. Takrat so spletni iskalniki dajali prednost stranem z največjo gostoto ključnih besed. To slabost iskalnikov je bilo lahko izigrati z večkratnim ponavljanjem ključnih besed in si tako zagotoviti visok položaj med iskalnimi rezultati. Larry Page in Sergey Brin sta prišla na idejo, da bi bile spletne strani na svetovnem spletu urejene hierarhično, po pomembnosti. Stran naj bi bila pomembna glede na število povezav, ki kažejo nanjo. Prvi dokument, ki opisuje delovanje algoritma PageRank, in njegova izvedba kot prototip v iskalniku Google, je izšel leta 1998. Kmalu za tem sta Page in Brin ustanovila danes veliko in znano podjetje Google. PageRank je danes le eden izmed dejavnikov, ki vplivajo na razvrstitev rezultatov pri iskanju.

2.2.2 O algoritmu PageRank

PageRank se uporablja za analizo povezav, ki ga je za svoje delovanje uporabljal spletni iskalnik Google. PageRank pripiše številčno vrednost vsaki strani svetovnega spleta, na katero naletijo Googlovi iskalni algoritmi. Dodeljena številčna vrednost predstavlja pomembnost spletne strani v razponu med 0 in 10. Čim večja je dodeljena vrednost, tem bolj pomembna je spletna stran (Slika 2.3). Vsaka spletna stran ima na začetku PageRank vrednost 0, ki se povečuje z večjim številom povezav, ki kažejo nanjo. Stran na spletu urejajo ljudje in sklepamo, da ustvarjajo povezave na pomembne strani. Tako se pomembnejšim stranem z vsako novo povezavo PageRank vrednost zvišuje. Pomembnost spletne strani oziroma PageRank vrednosti povečujemo z večjim številom kvalitetnih vhodnih povezav, te pa pridobimo z ustvarjanjem vsebin, ki jih uporabniki radi berejo in delijo z drugimi uporabniki.



Slika 2.3: Pomembnost spletne strani na internetu.

Naloga algoritma PageRank je pomoč spletnemu iskalniku, da vrne uporabniku kar najbolj pomembne in relevantne rezultate iskane poizvedbe. Google uporablja preko 200 različnih dejavnikov za rangiranje rezultatov. Večino dejavnikov lahko razdelimo na dve kategoriji, in sicer na ustreznost in pomembnost. Ustreznost pomeni, da je poizvedba del indeksa, ki vsebuje ključne besede. Pomembnost pomeni razvrstitev spletne strani. Potek iskanja opišemo s postopkom [6]:

1. uporabnik vnese želeno poizvedbo,
2. iskalnik preišče svoj indeks po ključnih besedah v poizvedbi,
3. iskalnik izbere spletne strani, ki ustrezajo poizvedbi,
4. izbrane spletne strani se sortirajo s pomočjo algoritma (npr. PageRank),
5. rezultati se prikažejo uporabniku.

Treba je omeniti tudi, da ima vsaka domena in njene poddomene svojo PageRank vrednost. Tabela 2.1 prikazuje PageRank vrednost popularne slovenske strani z avtomobilskimi oglasi, in sicer *www.avto.net*, ter njeni poddomeni. Vrednosti PageRank smo dobili s spletne strani *www.prchecker.net*.

V splošnem ni nujno, da ima poddomena nižjo PageRank vrednost kot pa glavna domena.

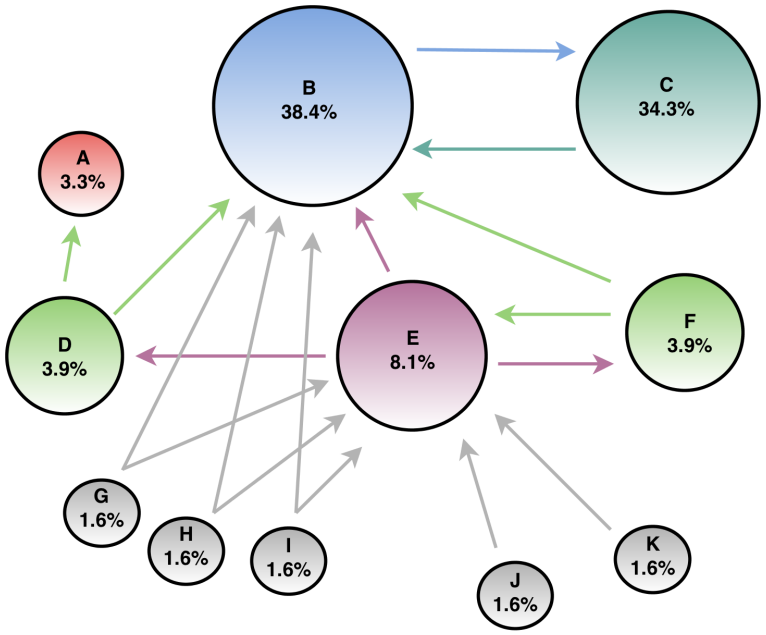
Domena/poddomena	PageRank vrednost
www.avto.net	5
www.avto.net/index.asp	4
www.avto.net/_AVTO/	3

Tabela 2.1: PageRank vrednosti spletnih strani.

2.2.3 Ideja algoritma PageRank

PageRank uporablja "glasovanje" za izračun svoje vrednosti. Spletna stran, ki vsebuje povezavo na drugo stran, ji odda en glas. Na PageRank vrednost ne vpliva samo število glasov, saj bi potem lahko lastniki spletnih strani naredili množico lažnih spletnih strani, ki bi kazale na njihovo, in tako na enostaven način prelisičili iskalni algoritem. Glavno vlogo igra pomembnost strani, ki oddaja glas. Če ima spletna stran povezavo iz zelo pomembne spletne strani (npr. PageRank vrednost 9), ima tak glas večjo vrednost kot deset povezav iz strani, ki imajo nižjo PageRank vrednost. Vsak glas je torej utežen.

Slika 2.4 prikazuje, kako glasovi iz pomembnih spletnih strani vplivajo na PageRank vrednost. PageRank vrednost je prikazana z vrednostmi med 0 in 100 zaradi lažje predstave. Lahko vidimo, da ima stran C večjo PageRank vrednost kot stran E, kljub temu, da ima manj vhodnih povezav. Vhodna povezava, ki kaže na stran C, je zelo pomembna, saj prihaja iz strani B, ki je najpomembnejša. Ta glas je vrednejši od vhodnih povezav na stran E, ki prihajajo večinoma od nepomembnih strani z zelo nizko PageRank vrednostjo.



Slika 2.4: Razporeditev PageRank glasov.

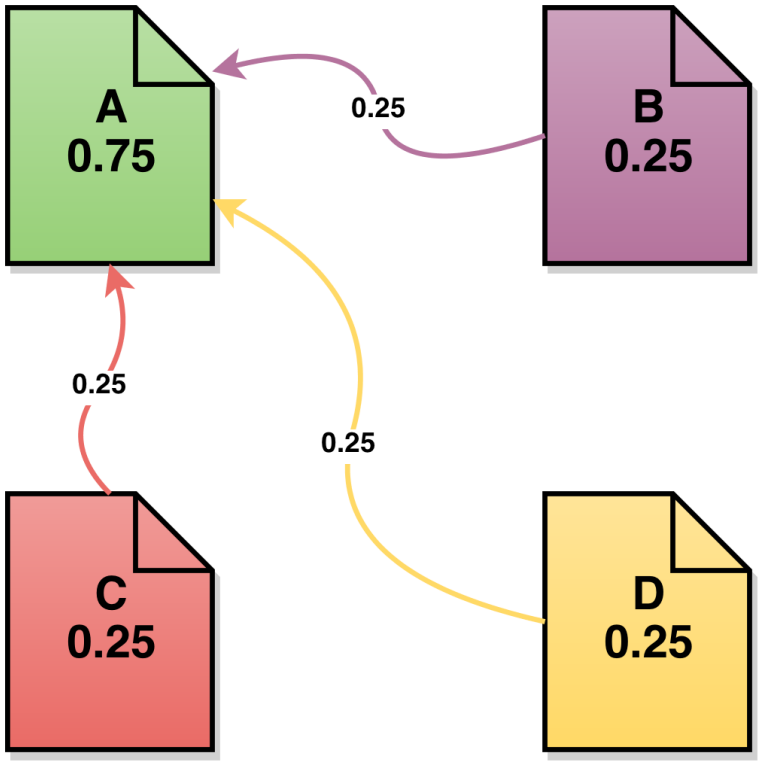
2.2.4 Poenostavljen algoritem za izračun PageRank vrednosti

Dosedaj smo opisali logiko in dejavnike, ki vplivajo na izračun PageRank vrednosti. V nadaljevanju poenostavljeno opišemo sam algoritem.

Predpostavimo, da imamo na spletu štiri spletne strani: A, B, C in D. Začetna aproksimacija PageRank vrednosti bi bila enako razporejena med vse štiri spletne strani. Tako bi vsaka spletna stran dobila začetno PageRank vrednost $0,25$. Če imajo strani B, C in D eno povezavo na stran A, ji prenesejo PageRank glas v vrednosti $0,25$. Ker vse povezave kažejo na stran A, dobi A seštevek glasov v vrednosti $0,75$ (Slika 2.5).

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

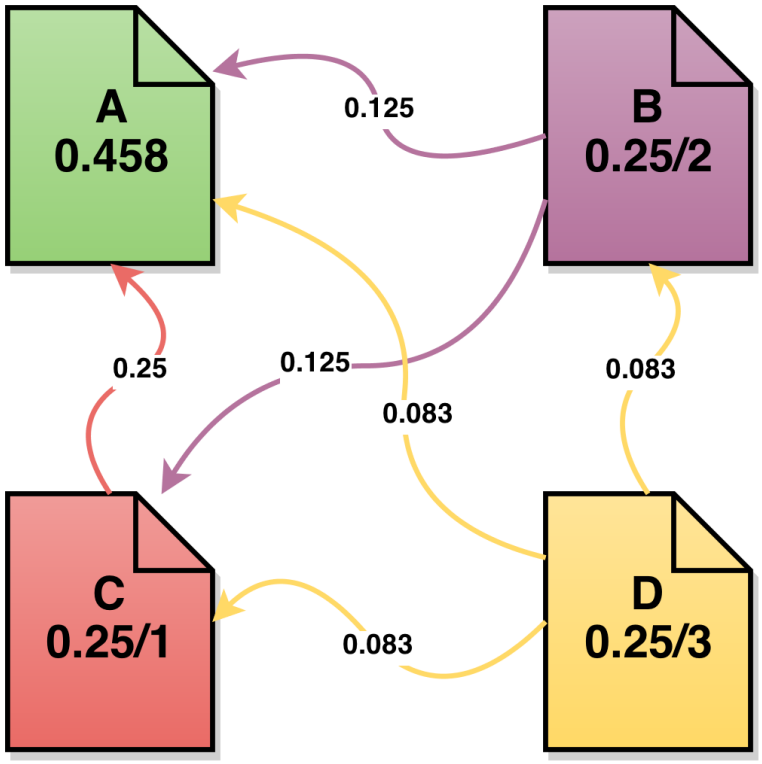
V naslednjem primeru ima stran B povezavo na strani A in C, stran C



Slika 2.5: Prikaz prenosa PageRank glasa (strani imajo eno povezavo). Prenese se celotna vrednost PageRank.

ima povezavo na A in stran D ima povezave na vse tri strani. PageRank vrednost posamezne strani se porazdeli med njene izhodne povezave. Tako stran B prenese polovico svoje vrednosti (0.125) strani A in drugo polovico $(0,125)$ strani C. Stran C prenese vso svojo vrednost $(0,25)$ na edino stran, na katero kaže, na stran A. Ker ima stran D tri izhodne povezave, prenese tretjino svoje vrednosti $(0,083)$ na stran A. Stran A ima na koncu PageRank vrednost $0,458$:

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



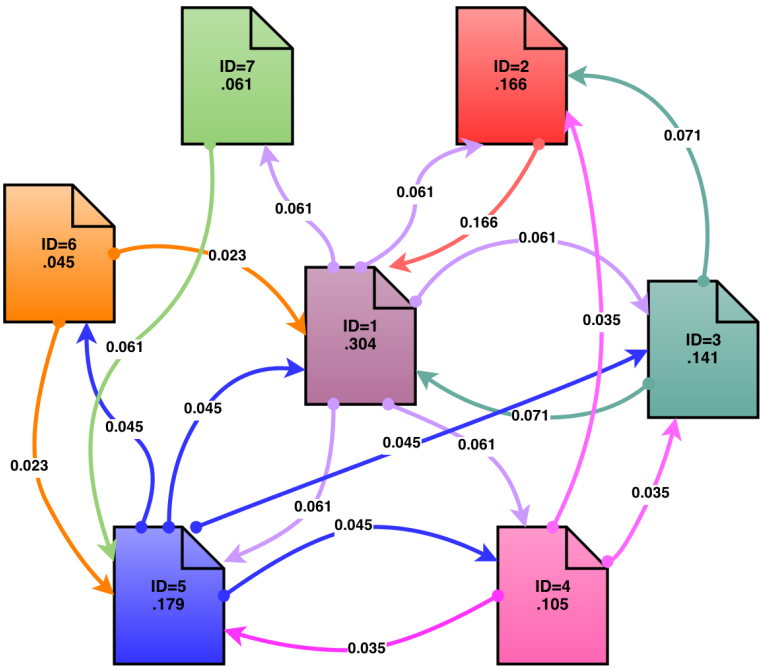
Slika 2.6: Prikaz prenosa PageRank glasa (strani imajo več povezav). Vrednost PageRank se deli s številom povezav.

PageRank vrednost se porazdeli med izhodne povezave oziroma PageRank vrednost strani se deli s številom izhodnih povezav L (Slika 2.6).

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

PageRank vrednost strani u, pri čemer odvisna od vrednosti vsake strani v, vsebovane v množici $M(u)$ (to je množica, ki vsebuje vse strani, ki kažejo na stran u), deljeno s številom njihovih povezav $L(v)$:

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



Slika 2.7: Algoritem PageRank v praksi.

Faktor dušenja

Algoritem PageRank upošteva tudi verjetnost, da bo naključni spletni uporabnik sledil verigi povezav na straneh in ne bo odšel na naključno stran. To verjetnost imenujemo faktor dušenja (d). Različne študije so testirale vrednosti faktorja dušenja. Za najbolj primerno se je izkazala (in se tako tudi največkrat uporablja) vrednost $0,85$. Poenostavljeno povedano:

- če je $d = 1$, PageRank predvideva, da uporabnik začne na naključni strani in potem s kliki na povezave obiskuje strani, ne da bi ročno vnesel naslov v brskalnik,
- če je $d = 0$, PageRank predvideva, da uporabnik nikoli ne klikne na povezavo in vedno ročno vnaša naslove v brskalnik.

Faktor dušenja algoritem najprej odšteje od 1 ter deli s številom vseh strani na svetovnem spletu. Ta vrednost se sešteje s produktom med faktorjem dušenja in vsoto PageRank vrednosti, ki so se prenesle z drugih strani. Za stran p_i dobimo:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)} \tag{2.1}$$

V različnih verzijah algoritma se uporabljata dve enačbi (2.1) in (2.2). Razlika med njima je, da se pri enačbi (2.1) faktor $(1-d)$ deli s številom N, ki predstavlja število vseh strani svetovnega spleta, pri enačbi (2.2) pa ne. V praksi se uporablja enačba (2.1), kot sta v svoji študiji zapisala Page in Brin, da je vsota vseh PageRank vrednosti enaka 1. V primeru druge enačbe pa je vsota vseh vrednosti enaka N.

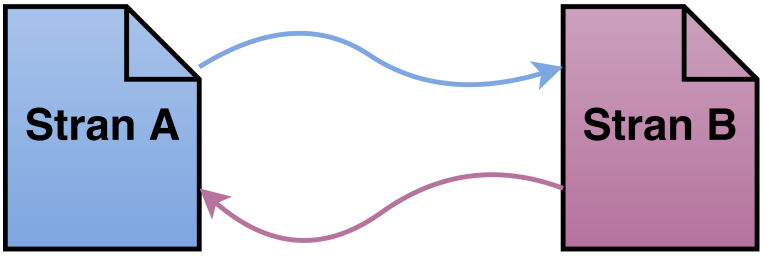
$$PR(p_i) = (1-d) + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)} \tag{2.2}$$

2.2.5 Primer izračuna PageRank vrednosti

Formula za izračun vrednosti PageRanka je iterativna. Izračun se ustavi, ko je razlika med vrednostjo predhodne iteracije in trenutne iteracije minimalna (npr. $0,0001$). Pokažimo, zakaj je potrebno za končno vrednost PageRank več iteracij. Slika 2.8 prikazuje primer, v katerem sta dve spletni strani povezani med seboj. Ker ne poznamo njunih začetnih PageRank vrednosti, predpostavimo, da ima vsaka stran vrednost 0.

Prva iteracija:

$$PR(A) = 0,15 + 0,85 * \frac{0}{1} = 0,15$$



Slika 2.8: Dve strani, povezanih med seboj.

$$PR(B) = 0,15 + 0,85 * \frac{0,15}{1} = 0,2775$$

V prvi iteraciji smo pri strani A upoštevali vhodno povezavo s strani B in začetno PageRank vrednost strani B. Pri strani B smo že upoštevali novo PageRank vrednost strani A. Ocena strani B še ni točna zaradi predvidevanja začetne vrednosti strani B pri izračunu vrednosti pri strani A.

Druga iteracija:

$$PR(A) = 0,15 + 0,85 * \frac{0,2775}{1} = 0,38587$$
$$PR(B) = 0,15 + 0,85 * \frac{0,38587}{1} = 0,47799$$

Tretja iteracija:

$$PR(A) = 0,15 + 0,85 * \frac{0,47799}{1} = 0,55629$$
$$PR(B) = 0,15 + 0,85 * \frac{0,55629}{1} = 0,62285$$

Četrta iteracija:

$$PR(A) = 0,15 + 0,85 * \frac{0,62285}{1} = 0,67942$$
$$PR(B) = 0,15 + 0,85 * \frac{0,67942}{1} = 0,72750$$

Po prvih iteracijah je vrednost PageRank še zelo netočna. Zato je treba ponavljati iteracije toliko časa, da vrednosti konvergirajo [6].

2.2.6 Slabosti in izigravanje PageRank algoritma

V želji po zvišanju PageRank vrednosti spletne strani in višji poziciji med rezultati iskanja nekateri lastniki spletnih strani poskušajo goljufati. Najbolj znana načina sta farme povezav (angl. link farms) in kupovanje povezav od pomembnejših spletnih strani.

Farma povezav je skupek spletišč (uporablja se preko 100 domen), v katerih se nahaja veliko, ponavadi lažnih, spletnih strani. Strani v takšnih spletiščih so močno povezane med seboj. Na ta način skušajo preslepiti algoritem, da z visoko oceno oceni stran, ki jo izberejo za najpomembnejšo. Na to stran imajo povezave prav vse strani, ta stran pa nima povezave na nobeno. Iz take strani naredijo povezavo na stran, ki ji želijo zvišati vrednost, in algoritem PageRank meni, da je stran zelo pomembna, saj nanjo kaže pomembna stran s celim glasom. Farme povezav je težko ločiti od realnih spletišč.

Google spletnim stranem, za katere odkrije, da so povezave dobile na nedovoljen način prek farm povezav ali prek kupovanja, dodeli zelo nizko vrednost. Algoritem PageRank je odpornejši na take goljufije kot starejše metode rangiranja, ki so štele število vhodnih povezav [2].

2.3 Google Penguin in Panda

Penguin in Panda sta novejša algoritma za rangiranje zadetkov. Google si z njima prizadeva izboljšati rezultate iskanj in uporabniško izkušnjo. Z njuno pomočjo izvaja posodobitve in kaznuje ali odstrani spletne strani, za katere se izkaže, da poskušajo na nedovoljen način priti do višje pozicije ali ustvarjajo slabo uporabniško izkušnjo. Hkrati pa izboljšujeta pozicijo spletnim stranem, ki upoštevajo Googlove napotke (angl. Google's Guidelines). Z upoštevanjem teh napotkov Google hitreje poišče, indeksira in rangira spletne strani. Če

Google ne bi redno izvajal posodobitev, bi rezultati iskanj vsebovali mnogo neželenih spletnih strani. Penguin je algoritem za iskanje takšnih strani. Kaznuje strani, ki so povezave pridobile prek farm povezav ali prek kupovanja povezav s pomembnejših spletnih strani. Panda poišče nekvalitetne spletne strani. Primer nekvalitetne strani bi bila stran, na kateri se pojavlja mnogo reklamnih sporočil ali pa nam klik na njo povzroči avtomatsko odpiranje neželenih reklamnih sporočil (angl. pop-up messages). Google je posodobitve nazadnje izvajal maja (Penguin) in julija (Panda) letos [19].

2.4 TrustRank in SandBox

TrustRank in SandBox sta del filtrov, ki jih Google uporablja za pozicioniranje spletnih strani.

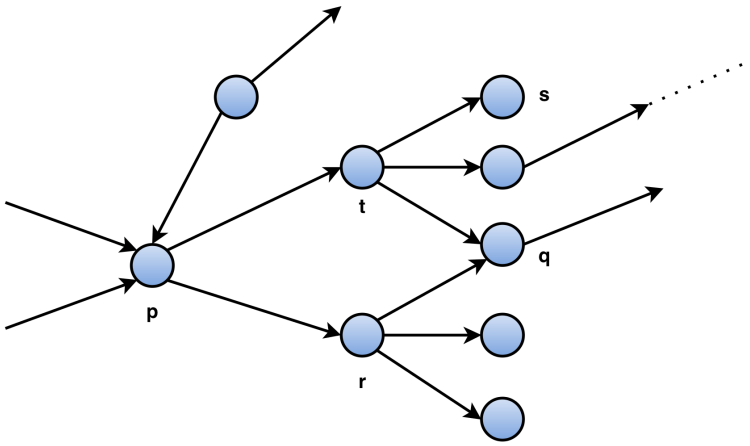
TrustRank je stopnja zaupanja, ki je dodeljena vsaki spletni strani na internetu in pove spletnim iskalnikom, koliko lahko zaupajo določeni spletni strani. Visok TrustRank pomeni, da spletni iskalnik zelo zaupa neki strani. S tem si stran zagotovi tudi visok položaj v rezultatih iskanj. Pri tem je ključnega pomena starost spletne strani. Starejša kot je stran, bolj ji spletni iskalniki zaupajo. Pomembna je tudi starost strani, iz katere prihajajo vhodne povezave, da pomembne strani vsebujejo povezavo na to stran, zgodovina pridobivanja vhodnih povezav (kdaj in kako hitro so bile pridobljene) in da v preteklosti ni pridobivala povezav na nedovoljen način [9].

SandBox si lahko predstavljamo kot preizkuševališče, v katerem se nahajajo nove in manj verodostojne strani za določen čas, da se prepreči pojavljanje nezaželenih naslovov (angl. spam) v rezultatih iskanj. Je testno okolje, kjer se hranijo nove spletne strani, dokler ne dokažejo, da so vredne zaupanja in ustrezajo pogojem, da jih spletni iskalnik prikaže med svojimi zadetki. Tja se premaknejo in se s tem se ne pojavljajo v rezultatih iskanj nove strani, za katere se izkaže, da so sledile slabi optimizacijski strategiji. To pomeni poskus hitrega pridobivanja zunanjih povezav v kratkem času, kar ponavadi pomeni, da so bile pridobljene na nedovoljen način (npr. prek

farm povezav). Strani v preizkušališču se lahko sčasoma premaknejo iz njega z grajenjem kvalitetnih, ustreznih povezav in uporabnikom prijazno vsebino [10].

2.5 DistanceRank

Iskalniki poleg algoritma PageRank uporabljajo tudi algoritem DistanceRank. Algoritem temelji na razdalji med spletnimi stranmi. Večja kot je povprečna razdalja med dvema stranema, večji je faktor kaznovanja oziroma manj pomembna je stran. Za razumevanje algoritma podajamo več definicij.



Slika 2.9: Logaritmična razdalja med p in q je enaka $log(2) + log(3)$.

Definicija 1
Minimalna razdalja med stranema i in j je definirana kot pot, po kateri uporabnik porabi najmanj klikov, da iz strani i pride do strani j.

Definicija 2
Če ima stran i povezavo na stran j, potem je teža te povezave med stranema i in j enaka $log_{10}O(i)$, kjer $O(i)$ predstavlja število izhodnih povezav s strani i.

Definicija 3
Razdalja med stranema i in j je enaka teži najkrajše poti (pot z minimalno vrednostjo) med stranema i in j. Tej razdalji pravimo logaritmična razdalja in jo označujemo z d_{ij}.

Slika 2.7 prikazuje, da so teže izhodnih povezav spletnih strani p, r in t enake $log(2)$, $log(4)$ in $log(3)$. Pot od p do q ima razdaljo enako $log(2) + log(3)$, če je pot p–r–q najkrajša pot od p do q. Pot od p do s pa je enaka $log(2) + log(4)$. Obe strani q in s sta od strani p oddaljeni enako (dva klika), vendar je stran q bližje strani p, ker ima pot $log(2)+log(3)$ krajšo logaritmično razdaljo (minimalno težo povezav).

Definicija 4
Če d_{ij} predstavlja logaritmično razdaljo med stranema i in j (po definiciji 3), potem d_j predstavlja povprečno razdaljo strani j ali pomembnost strani j,

$$d_j = \frac{\sum_{i=1}^{V} d_{ij}}{V}$$

pri čemer V označuje število vseh spletnih strani.

Cilj algoritma je zmanjšati faktor kaznovanja ali razdaljo z namenom, da imajo strani s krajšo razdaljo višjo oceno. Rezultati kažejo, da je DistanceRank celo boljši in njegovo delovanje manj zapleteno od drugih razvrstitvenih algoritmov. Na končni rezultat ne vplivajo strani, ki nimajo vhodnih oz. izhodnih povezav, kot to velja pri algoritmu PageRank. Prav tako ni treba upoštevati faktorjev, kot je npr. faktor dušenja pri algoritmu PageRank, saj se DistanceRank vedno izvaja na realnih grafih. Slabost je le kompleksnost logaritmične razdalje, ki znaša $O(|V|*|E|)$ (pri algoritmu PageRank je to najslabši scenarij, ponavadi je za sprejemljiv rezultat dovolj že 100 ponovitev), kar je pri 11,5 milijard strani na spletu zelo veliko [5].

Poglavje 3

Uporabljena orodja in tehnologije

V tem poglavju na kratko opišemo orodja in tehnologije, ki smo jih uporabili pri razvoju spletne aplikacije.

3.1 Razvojno okolje NetBeans IDE

NetBeans [13] je integrirano razvojno okolje (IDE). Prvotno je bilo razvito za razvoj aplikacij in programske opreme v programskem jeziku Java, vendar je sedaj omogočen razvoj tudi v drugih programskih jezikih, kot so C/C++, XML, HTML, PHP, Groovy, Javadoc, JavaScript in JSP. NetBeans IDE je napisan v Javi in deluje na operacijskih sistemih Windows, OS X, Linux, Solaris in drugih platformah, ki podpirajo JVM (Java Virtual Machine). Verzija NetBeans IDE 7.3, ki je izšla februarja 2013, je dodana podpora za HTML5 in razvoj spletnih tehnologij. Za razvoj spletnih aplikacij je treba namestiti HTML5 Web Development Support, ki vsebuje [14]:

- ustvarjanje projektov v HTML5,
- predogled spletne strani,
- urejevalnik za HTML, CSS in JavaScript,

- razhroščevalnik (angl. debugger) za JavaScript.

3.2 HTML

HTML (HyperText Markup Language) je označevalni jezik za oblikovanje in prikazovanje spletnih strani in drugih vsebin v spletnem brskalniku. Vsebuje HTML elemente, sestavljene iz značk (angl. tags). Značke se vedno nahajajo med znakoma < in >, kot naprimer <html>. HTML značke najpogosteje nastopajo v obliki parov, npr. <h1> in </h1>. Nekatere značke predstavljajo prazne elemente in so brez para, npr. <img>. Prvo značko v paru imenujemo začetna značka, drugo pa končna značka. Mednju lahko dodamo besedilo, dodatne značke, komentarje in druge vrste tekstovnih vsebin.

Spletni brskalniki preberejo HTML dokumente in jih pretvorijo v vidno in slišno obliko. Spletni brskalnik ne prikazuje HTML značk, uporablja jih za prikaz in oblikovanje vsebine spletne strani.

HTML dovoljuje uporabo slik in drugih objektov na spletni strani, uporablja pa se tudi za kreiranje spletnih obrazcev. Z njim lahko oblikujemo strukturirane dokumente z označevanjem semantike besedila, kot so naslovi, odstavki, seznami, povezave, citati itd. V HTML dokumente lahko vključimo tudi kodo (npr. JavaScript), ki vpliva na prikaz in funkcionalnost spletne strani [15].

3.3 HTML5

HTML5 (HyperText Markup Language, verzija 5) je naslednik trenutnega standarda HTML. Tako kot njegova predhodnika, HTML 4.01 in XHTML 1.1, je HTML5 standard za oblikovanje in predstavitev vsebin na svetovnem spletu. Nastaja s ciljem, da poenostavi sintakso predhodnikov in prinese vanjo nekaj strukturne urejenosti XML-ja (Extensible Markup Language).

Želja po interaktivnih vsebinah in aplikacijah s širokim naborom funkcij je vodila razvijalce brskalnikov in razvijalce aplikacij do pogoste uporabe do-

datnih vtičnikov. Dva najpogostejša vtičnika sta ActiveX (Microsoftov okvir za definiranje programskih komponent) in Flash (Adobovo orodje za interaktivne vsebine). Dosedanja uporaba vtičnikov je zahtevala več procesorske zmogljivosti, dodatno kodiranje s strani razvijalca aplikacij in povzročala počasnejše nalaganje spletnih strani. HTML5 ponuja širok nabor elementov, s katerimi se je marsikdaj mogoče izogniti dodatnim tehnologijam.

HTML5 prinaša enostavnost. Grajen je bil z namenom, da obstoječe strani delujejo še naprej tako kot sedaj, novim pa ponudi moč novih atributov. V HTML5 delujejo vsi elementi iz predhodnikov (HTML4.01 in XHTML1.1) pravilno, čeprav so nekateri elementi v specifikaciji HTML5 registrirani kot zastareli in naj jih ne bi več uporabljali [4].

HTML5 uvaja številne nove elemente, npr. <video>, <audio> in <canvas> elemente. Novi elementi, kot so <section>, <article>, <header> in <nav>, so namenjeni za obogatitev vsebine dokumentov. Nekateri novi atributi nadomeščajo stare. Elementi, kot so <a>, <cite> in <menu>, so bili spremenjeni in standardizirani. Aplikacije in objekti elementov so temeljni del specifikacije HTML5. Osnovne razlike HTML5 od HTML4 so [16]:

- nova pravila uporabe elementov,
- novi elementi: header, footer, section, article, video, audio, progress, nav, meter, time, aside, canvas,
- novi tipi vhodnih elementov: color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url, week,
- novi atributi,
- video posnetke in glasbo lahko predvajamo kar na spletni strani.

3.4 CSS

CSS (Cascading Style Sheets) so predloge v obliki slogovnega jezika, ki skrbi za izgled spletnih strani. Z njimi definiramo stil HTML oz. XHTML ele-

mentov v smislu pravil, kako naj se ti prikažejo na spletni strani. Določamo lahko barve, velikosti, odmike, poravnave, obrobe, pozicije in vrsto drugih atributov, prav tako pa lahko nadziramo uporabnikove aktivnosti, ki jih izvaja nad elementi, vključenimi na spletno stran. Bistvo uporabe CSS je ločevanje vsebine od izgleda dokumenta. S tem omogočimo lažje urejanje in dodajanje stilov ter poskrbimo za večjo preglednost dokumentov, ki temeljijo na HTML sintaksi. Prav tako zmanjšamo ponavljanje kode, saj omogočimo množici strani uporabo istih podlog, kar lahko bistveno zmanjša velikost dokumentov. CSS omogoča tudi zlivanje oz. prepisovanje pravil iz več virov, kar imenujemo kaskadiranje. Pravila z višjo prioriteto bodo prepisala tiste z nižjo [17].

3.5 JavaScript

JavaScript je objektno orientiran skriptni programski jezik, ki ga je razvil Netscape, v pomoč pri ustvarjanju interaktivnih spletnih strani. Jezik je bil razvit neodvisno od Jave, vendar si z njo deli številne lastnosti in strukture. Uporablja se v kombinaciji s HTML kodo in s tem poživi stran z bolj dinamičnim izvajanjem. JavaScript se uporablja za razvijanje dinamičnih spletnih strani. Program, napisan v JavaScriptu, se vključi ali vgradi direktno v HTML kodo. S tem je možno izvajati funkcije in naloge, kot so npr. enostavni izračuni ali preverjanje pravilnosti vnesenih podatkov. Na žalost je za podporo vseh brskalnikov treba implementirati več različic funkcij, ker različni spletni brskalniki uporabljajo različne objekte. Zunaj spleta se JavaScript uporablja v različnih orodjih. Adobe Acrobat in Adobe Reader ga podpirata v datotekah PDF. Podpirata ga tudi operacijska sistema Microsoft Windows in Mac OS X [18].

Poglavje 4

Spletna aplikacija

4.1 Razvoj spletne aplikacije

4.1.1 Implementacija algoritma PageRank

Razvoj spletne aplikacije smo začeli z izvedbo algoritma PageRank v programskem jeziku Java. Kasneje smo razvoj algoritma prenesli v programski jezik JavaScript, ki je namenjen razvoju spletnih aplikacij. Spletne strani in njihove povezave hranimo v podatkovni strukturi razpršena tabela (angl. *hash table*, tudi *hash map*). Razpršena tabela je podatkovna struktura, v kateri so podatki dosegljivi po ključih (angl. *keys*), vsebujejo pa vrednosti (angl. *values*). V našem primeru smo spletne strani shranili kot ključe, njihove povezave pa kot vrednosti (Tabela 4.1). Na začetku imajo spletne strani enake PageRank vrednosti (vsota vseh strani 1 se deli s številom vseh strani). V naslednjem koraku smo preko zanke iz tabel povezav, ki jo ima vsaka spletna stran shranjeno v razpršeni tabeli, dobili posamezne povezave in stranem, na katere povezave kažejo, prenesli vrednost PageRank. Prenešena vrednost je odvisna od trenutne vrednosti PageRank spletne strani in števila izhodnih povezav.

Razpršena tabela	
Ključ (spletna stran – tip Integer)	Vrednost (seznam povezav – tip Array())
0	6, 2, 10, 8, 1, 15, 3
1	12, 4, 8, 2, 13, 18, 16, 5, 6, 14, 21
2	8, 18
...	...
N	1, 15, 25

Tabela 4.1: Primer uporabe razpršene tabele za hranjenje spletnih strani in njihovih povezav.

Nekatere spletne strani so brez izhodnih povezav. Problem takih strani je, da naključni uporabnik (angl. random surfer) obstane na takih straneh. Njihov del vrednosti PageRank se ne prenaša, kar ni pravično do drugih strani, ki svojo vrednost prenašajo na druge. Zato vrednosti vseh strani brez povezav seštevamo in na koncu delimo s številom vseh spletnih strani. Ta vrednost se vsaki strani prišteje k njeni vrednosti PageRank. V zadnjem koraku se pri vrednostih PageRank spletnih strani upošteva faktor dušenja.

Ta postopek se ponavlja toliko časa, dokler razlika PageRank vrednosti predhodne iteracije in trenutne iteracije ni manjša od $0,0001$. Postopek je predstavljen s psevdo kodo (Slika 4.1).

4.1.2 Implementacija spletne aplikacije

Glavni del spletne aplikacije je HTML5 element *canvas*. Je ena izmed najbolj uporabnih novosti HTML5. Pri spletni aplikaciji smo ga uporabili v kombinaciji s programskim jezikom JavaScript za risanje in prikazovanje dvodimenzionalne grafike. Za delo z miško smo na elementu *canvas* definirali dogodke nad njo. Za potrebe aplikacije potrebujemo štiri dogodke, in sicer "*mousemove*", "*mousedown*", "*mouseup*" in "*dblclick*". Dogodek "*mousemove*" se uporablja za pridobivanje koordinat miškinega kurzorja, "*mousedown*" in "*mouseup*" potrebujemo za konec oz. začetek premikanja

objektov po elementu *canvas* in "*dblclick*" za dodajanje povezav oz. izbris le-teh. Dogodek "*mousedown*" ima poleg premikanja objektov po prostoru, še funkcijo dodajanja novih spletnih strani, če se miškin klik zgodi na praznem prostoru. Spletne strani so definirane kot objekti. Vsaka stran je en objekt, ki ima več atributov:

- koordinati x in y,
- r označuje premer kroga, ki predstavlja spletno stran,
- atribut *mouse*, ki pove, če je miškina pozicija nad objektom,
- atribut *drag*, ki pove, če se element premika,
- *text* vsebuje ime objekta,
- n je zaporedna številka objekta,
- *rank* je vrednost PageRank objekta,
- *connections* vsebuje izhodne povezave objekta,
- *lineCoordinatesStart* in *lineCoordinatesEnd* se uporabljata pri izbrisu povezav,
- *color* predstavlja barvo objekta.

S klikom na prazen prostor se kreira nov objekt s prej naštetimi atributi. Objekte hranimo v tabeli. Z vsakim novim objektom se kliče funkcija za izračun vrednosti PageRank. Ob klicu funkcija najprej iz drsnika prebere vrednost faktorja dušenja. Poleg tega se ta funkcija kliče tudi ob dodajanju ali brisanju povezav in brisanju celotnih objektov. Pri izračunu vrednosti PageRank se hkrati v posebno tabelo shranjujejo vrednosti PageRank spletnih strani po iteracijah. To tabelo uporablja drsnik za iteracije, s katerim na aplikaciji opazujemo, kako so se vrednosti PageRank obnašale po iteracijah, in gumb, s katerim prikažemo animacijo. Pri vsaki operaciji nad objekti se sproži funkcija za izris objektov na *canvas* oz. zaslon.

procedure PageRank(G) ▷ G: seznam strani in njihovih povezav
  $dF \leftarrow 0,85$ ▷ faktor dušenja: 0.85
  $izh \leftarrow G$ ▷ število izhodnih povezav v G
  $vh \leftarrow G$ ▷ število vhodnih povezav v G
  $N \leftarrow G$ ▷ število spletnih strani v G
  $razlika \leftarrow 0$
  $ite \leftarrow 0$ ▷ število iteracij
  for vsak s v grafu **do**
    $PR[s] \leftarrow \frac{1}{N}$ ▷ Inicializeramo začetne vrednosti PageRank
  end for
  while razlika $> 0,001$ **do**
    $pPR \leftarrow PR$ ▷ Shranimo vrednosti PageRank iz prejšnje iteracije
    $ite \leftarrow ite + 1$
    $dp \leftarrow 0$
    if vsak s brez izhodnih povezav **then**
      $dp \leftarrow dp + \frac{PR[s]}{N}$
    end if
    for vsak ip v $vh[s]$ **do**
      $pomPG[s] \leftarrow pomPG[s] + \frac{PR[ip]}{iz[ip]}$ ▷ Prenos vrednosti PageRank
    end for
    for vsak s v grafu **do**
      $pomPG[s] \leftarrow \frac{1-d}{N} + dF * \frac{PR[s]}{iz[s]}$ ▷ upoštevanje faktorja dušenja
    end for
    $PR \leftarrow pomPR$ ▷ Posodobitev PageRank
    for vsak s v grafu **do**
      $razlika \leftarrow razlika + |\frac{PR[s]}{iz[s]} - \frac{pPR[s]}{iz[s]}|$
    end for
  end while
end procedure

Slika 4.1: Psevdo koda algoritma PageRank.

4.2 Predstavitev spletne aplikacije

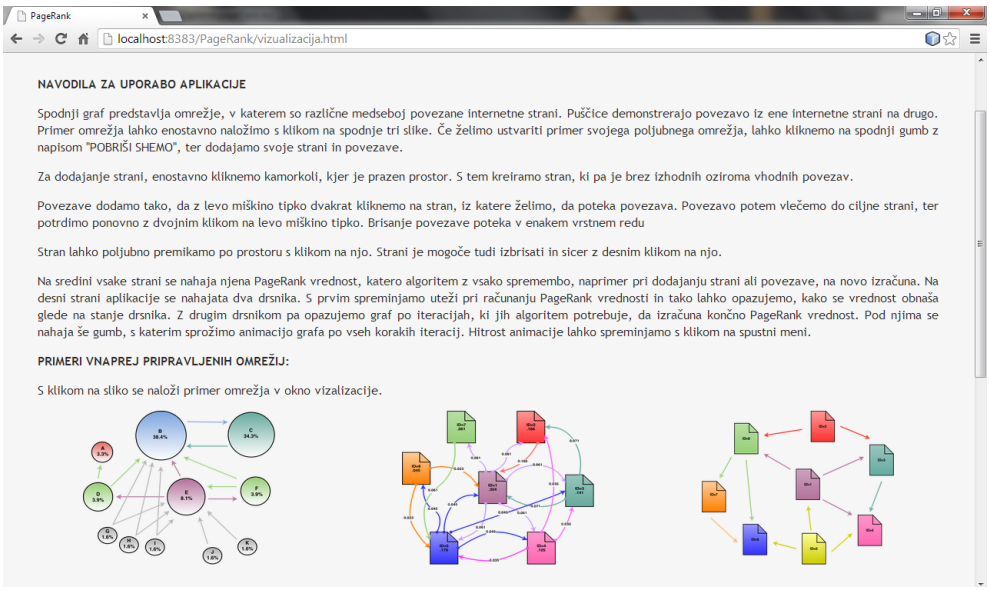
V nadaljevanju s slikami opišemo sestavo spletne strani in funkcionalnost drsnikov in gumbov. Za predstavitev spletne aplikacije smo uporabili spletni brskalnik Chrome. Chrome podpira vse uporabljene elemente HTML5 in CSS3, ki uradno še nista standarda, zato smo pri ostalih brskalnikih naleteli na težave pri prikazovanju elementov. Težave so se pojavile pri prikazu drsnikov in tudi pri sami vizualizaciji, kjer se grafi, ki prikazujejo spletne strani in povezave, niso izrisali.

Slika 4.2 prikazuje glavno stran spletne aplikacije. Na vrhu se poleg naslova nahaja meni, preko katerega dostopamo do glavne strani in strani, kjer je prikazana vizualizacija. Na glavni strani se nahaja besedilo o algoritmu PageRank, kjer je predstavljena zgodovina in delovanje algoritma.

Slika 4.2: Glavna stran spletne aplikacije "index.html".

Stran, kjer se nahaja vizualizacija algoritma PageRank, je prikazana na Sliki 4.3. Na začetku strani so navodila za uporabo spletne aplikacije. Vsebujejo postopeke za nalaganje vnaprej pripravljenih primerov grafov, dodajanje in izbris spletnih strani oz. povezav in funkcije gumbov in drsnikov. S slikami

so v osrednjem delu strani prikazani primeri vnaprej pripravljenih grafov. S klikom na sliko izberemo graf, nad katerim bi radi izvajali vizualizacijo.

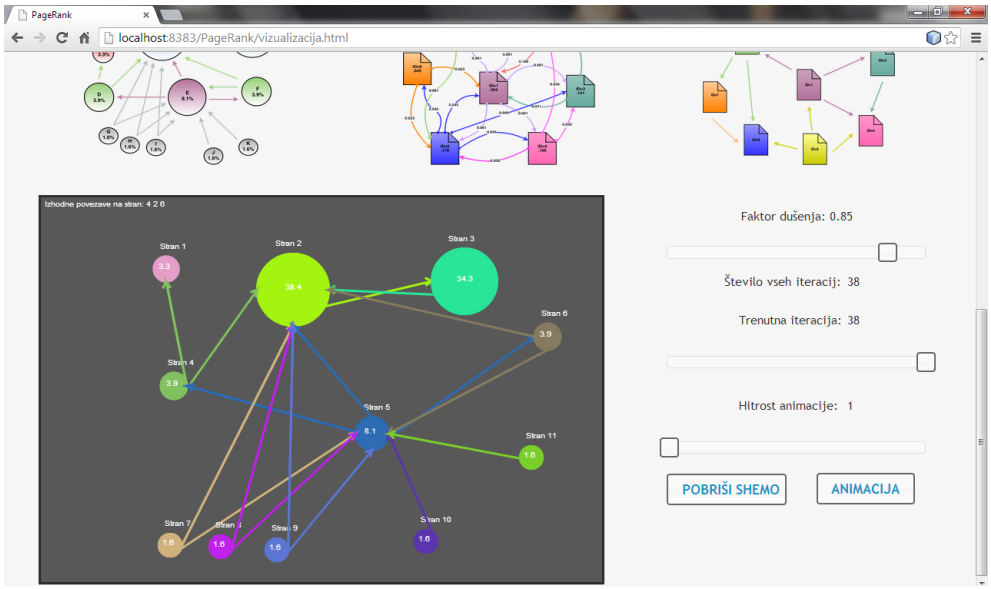


Slika 4.3: Stran z vizualizacijo, kjer se nahajajo navodila za uporabo spletne aplikacije in sama aplikacija.

Na spodnjem, levem delu strani je vizualizacija algoritma PageRank. Na desnem delu so drsniki, s katerimi nastavljamo:

- faktor dušenja,
- iteracije,
- hitrost animacije.

Nad vsakim drsnikom je napis, ki prikazuje trenutno stanje na drsniku. Pod drsniki se nahajata gumba "Pobriši shemo" in "Animacija". S pritiskom na gumb "Pobriši shemo" pobrišemo zgrajeni graf, pritisk na gumb "Animacija" sproži animacijo, ki prikaže konvergiranje vrednosti PageRank vozlišča pri ponovitvah.



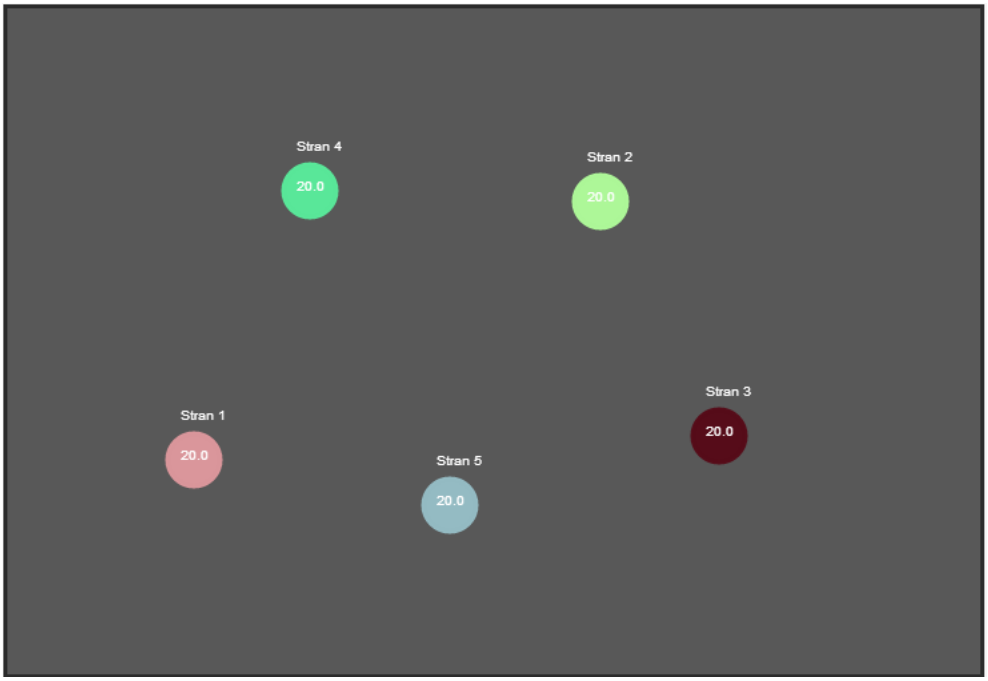
Slika 4.4: Prostor na strani, kjer je prikazana vizualizacija in prostor z drsniki ter dvema gumboma.

Vizualizacija

Vizualizacija je prikazana s pomočjo HTML5 elementa *canvas*. Spletne strani so v vizualizaciji predstavljene kot vozlišča. Za lažje ločevanje povezav med vozlišči so ta obarvana z različnimi barvami. Velikost vozlišča predstavlja njegovo PageRank vrednost, ki pa je prikazana tudi kot številčna vrednost na sredini vozlišč. Vrednosti so v razponu med 0 in 100 zaradi lažje demonstracije.

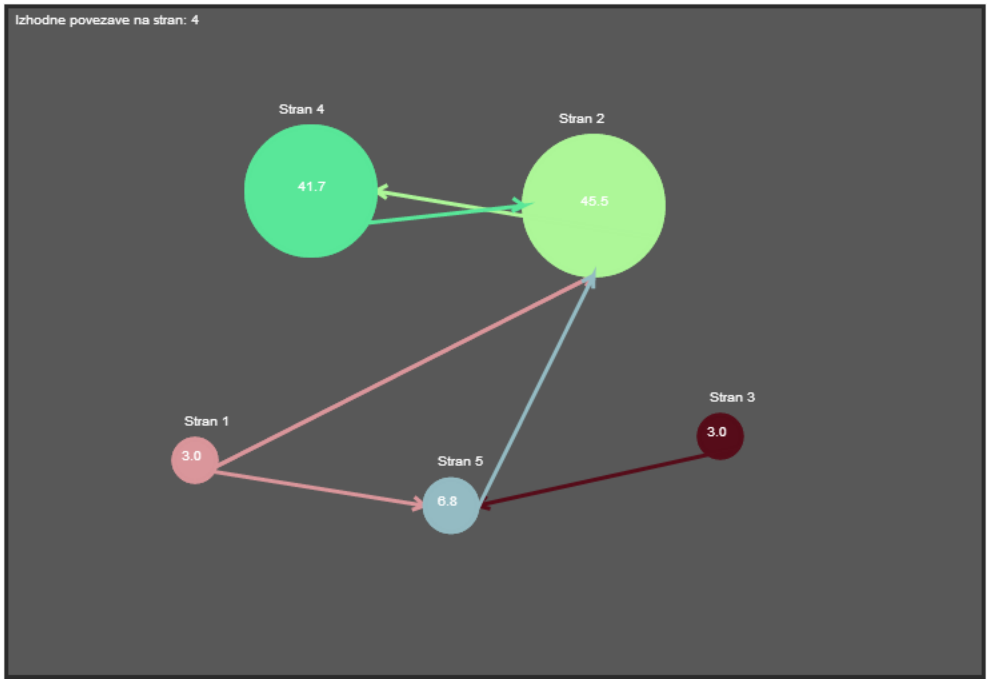
Spletne strani dodajamo s pritiskom na levo miškino tipko kamorkoli na prazen prostor. Ko je stran dodana, jo lahko z istim postopkom premikamo po prostoru. Če želimo izbrisati posamezno spletno stran, to storimo s pritiskom na desno miškino tipko. Če je graf brez povezav, imajo vsa vozlišča enako PageRank vrednost, ker se vrednost 1 (kar predstavlja vsoto vseh vozlišč) porazdeli po vozliščih (Slika 4.5).

Šele z dodajanjem povezav med vozlišči se spletne strani začnejo razvrščati po pomembnosti (Slika 4.6). Povezave dodajamo tako, da z levo miškino



Slika 4.5: PageRank vrednost spletnih strani brez povezav.

tipko dvakrat kliknemo na spletno stran, iz katere želimo povezavo. Povezavo nato vlečemo do ciljne spletne strani in jo potrdimo s ponovnim dvakratnim klikom na levo miškino tipko. Poleg puščic, ki ponazarjajo smer oddaje glasov, je to mogoče spremljati tudi v zgornjem levem kotu. Napis se pojavi vsakič, ko se z miškinim kurzorjem premaknemo na spletno stran.



Slika 4.6: PageRank vrednost spletnih strani s povezavami.

Poglavje 5

Zaključek

V diplomskem delu smo razvili spletno aplikacijo za vizualizacijo algoritma PageRank. Predstavili smo zgodovino ter delovanje algoritma PageRank na poenostavljenem primeru. Algoritem smo demonstrirali na praktičnem primeru in opisali njegove slabosti. Omenjamo tudi druge načine za rangiranje spletnih strani. Predstavimo implementacijo algoritma in spletne aplikacije ter podamo psevdo kodo algoritma PageRank. Na kratko opišemo uporabljene tehnologije in orodja. S slikovnim gradivom predstavimo spletno aplikacijo.

Cilj diplomskega dela je bil predstaviti algoritem PageRank. Vizualizirali in predstavili smo ga v spletni aplikaciji, ki uporabniku pomaga razumeti delovanje algoritma. Aplikacija ponuja izbiro vnaprej pripravljenih primerov grafov, uporabnik pa ima tudi možnost, da ustvari svoje primere.

Ideja za nadaljnje delo je izdelava vizualizacije in animacije še za ostale metode rangiranja spletnih strani in to vključiti v spletno stran. Smiselno bi bilo omeniti tudi vse nasvete in trike za rangiranje oz. optimizacijo spletnih strani. Tako bi uporabnik na enem mestu pridobil informacije, ki so koristne za postavitev in vzdrževanje dobre spletne strani.

Slike

Tabele

Literatura

[1] Tajner T., Optimizacija spletnih strani v centrih za izobraževanje odraslih, diplomsko delo, Pedagoška fakulteta Univerze v Ljubljani, 2012. Dostopno na:
http://pefprints.pef.uni-lj.si/910/1/TAJNER.pdf

[2] Tauzes V., Spletni iskalniki in PageRank, 2008. Dostopno na:
http://ibmi.mf.uni-lj.si/jure/pred_bib/ivi/seminarji-08/PageRank.pdf

[3] Stanko B., Optimizacija spletnih strani, diplomsko delo, Fakulteta za računalništvo in informatiko Univerze v Ljubljani, 2008. Dostopno na:
http://eprints.fri.uni-lj.si/609/1/StankoB_VS.pdfi

[4] Mišo Krog, Osnovne novosti v HTML5 in CSS3, članek, Pedagoška fakulteta Koper, 2011. Dostopno na:
http://90.157.203.179/clanek_HTML5_in_CSS3_Miso_Krog.pdf

[5] Ali Mohammad Zareh Bidoki, Nasser Yazdani, DistanceRank: An intelligent ranking algorithm for web pages, članek, Department of Electrical and Computer Engineering, University of Tehran, ScienceDirect, 2007. Dostopno na:
http://goanna.cs.rmit.edu.au/ aht/tiger/DistanceRank.pdf

[6] Gvajc B., Optimizacija spletnih strani za spletne iskalnike, diplomsko delo, Fakulteta za računalništvo in informatiko Univerze v Ljubljani,

2011. Dostopno na:
http://eprints.fri.uni-lj.si/1350/1/Gvajc1.pdf

[7] Romac G., Organizacija telekomunikacijske mreže: Google PageRank, seminar, Zavod za telekomunikacije, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, 2009. Dostopno na:
https://www.fer.hr/_download/repository/Google_PageRank.pdf

[8] PageRank. Wikipedia. Dostopno na:
http://en.wikipedia.org/wiki/PageRank
(dostop: 15. junij 2013)

[9] PageRank vs. Trust Rank : Real SEO ranking factor. Dostopno na:
http://www.seohawk.com/blog/page-rank-trust-rank-seo/
(dostop: 5. avgust 2013)

[10] Google Sandbox Effect Explained. Dostopno na:
http://www.seosandwitch.com/2012/08/google-sandbox-effect-explained.html
(dostop: 5. avgust 2013)

[11] Optimizacija spletnih strani za iskalnike. Dostopno na:
http://optimizacija-za-iskalnike.studiostyle.si/optimizacija_za_iskalnike/page_rank.html
(dostop: 5. avgust 2013)

[12] Optimizacija spletnih strani. Wikipedia. Dostopno na:
http://sl.wikipedia.org/wiki/Optimizacija_spletnih_strani
(dostop: 12. september 2013)

[13] NetBeans IDE. Wikipedia. Dostopno na:
http://en.wikipedia.org/wiki/NetBeans#NetBeans_IDE_Bundle_for_Web_and_Java_EE
(dostop: 12. september 2013)

[14] HTML5 Web Development Support. Dostopno na:
https://netbeans.org/features/html5/
(dostop: 15.september 2013)

[15] HTML. Wikipedia. Dostopno na:
http://en.wikipedia.org/wiki/HTML
(dostop: 15. september 2013)

[16] Kaj je HTML 5. Dostopno na:
http://spletnisistemi.si/blog/2011/02/04/kaj-je-html-5/
(dostop: 15. september 2013)

[17] CSS. Wikipedia. Dostopno na:
http://sl.wikipedia.org/wiki/CSS
(dostop: 16. september 2013)

[18] JavaScript. Wikipedia. Dostopno na:
http://sl.wikipedia.org/wiki/JavaScript
(dostop: 16. september 2013)

[19] Google Penguin and Panda: A Simple Explanation. Dostopno na:
http://www.stlouisdigitalmedia.com/blog/local-seo/google-penguin-panda-a-simple-explanation/
(dostop: 16. september 2013)