Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

# Page Rank

Bakalárska práca

2018 Tatiana Jánošová

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

# Page Rank

Bakalárska práca

| | |
|---|---|
| Študijný program: | Matematika |
| Študijný odbor: | 1113 Matematika |
| Školiace pracovisko: | Katedra algebry, geometrie a didaktiky matematiky |
| Školiteľ: | doc. RNDr. Andrej Ferko, CSc. |

Bratislava, 2018 Tatiana Jánošová

13568267

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

# ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Tatiana Jánošová
**Študijný program:** matematika (Jednoodborové štúdium, bakalársky I. st., denná forma)
**Študijný odbor:** matematika
**Typ záverečnej práce:** bakalárska
**Jazyk záverečnej práce:** slovenský
**Sekundárny jazyk:** anglický

**Názov:** Page Rank (e-learning)
*Page Rank (e-learning)*

**Anotácia:** Na patente PageRank v spoločnosti Google založili vyhľadávací algoritmus. Pre študujúcich matematiky a informatiky v projekte vytvorime výučbovy portál o algoritme PageRank. Spracujeme výklad problému a riešenia mocninnou metódou a navrhneme adekvátnu vizualizáciu aj aktivizáciu.

**Cieľ:** Vytvoriť e-learning na podporu výučby algoritmu Page Rank (PR). Výklad histórie PR, základné pojmy, definície, riešenie úlohy mocninnou metodou a možnosti "oklamania" výsledkov. Návrh originálnych riešených príkladov a prezentácie vizualizačnej (web, video) a aktivizačnej (napr. kvíz, testy). Na implementaciu si zvolíme vhodný autorský nástroj (napr. Matlab, Moodle).

**Literatúra:** Page, L. et al. 1999. The PageRank citation ranking: Bringing order to the Web. Technical report. [online] http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf.
Amy N. Langville and Carl D. Meyer. 2006. Google's PageRank and Beyond: The Science of Search Engine Rankings. [online] http://geza.kzoo.edu/~erdi/patent/langvillebook.pdf. Princeton University Press 2006.

**Kľúčové slová:** Page Rank, information vizualization

**Vedúci:** doc. RNDr. Andrej Ferko, PhD.
**Katedra:** FMFI.KAGDM - Katedra algebry, geometrie a didaktiky matematiky
**Vedúci katedry:** doc. RNDr. Pavel Chalmovianský, PhD.

**Spôsob sprístupnenia elektronickej verzie práce:**
bez obmedzenia

**Dátum zadania:** 02.11.2017

**Dátum schválenia:** 16.11.2017

prof. RNDr. Ján Filo, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

**Poďakovanie:** Týmto by som chcela poďakovať školiteľovi mojej bakalárskej práce doc. RNDr. Andrejovi Ferkovi CSc. za jeho cenné rady, odborný dohľad a metodickú pomoc pri jej vypracovaní.

# Abstrakt

JÁNOŠOVÁ, Tatiana:*Page Rank.* [bakalárska práca]. Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra algebry, geometrie a didaktiky matematiky. Školiteľ práce: doc. RNDr. Andrej Ferko, CSc. Obhajoba: 2018, s.35

Bakalárska práca je zameraná na vysvetlenie a implementáciu algoritmu Page Rank. Primárnym cieľom je vysvetlenie matematického modelu algoritmu a následné spracovanie témy do E-learningu. Sekundárnym cieľom je ozrejmiť dôvod vzniku a oboznámiť čitateľa s inou alternatívou pre ohodnotenia stránok. Ďalej rozoberáme postup práce pri tvorbe vizualizačnej časti pozostávajúcej z webovej stránky a doplňujúcej aktivizačnej časti, ktorú tvorí funkcia Page Rank v Matlabe. Prvá polovica práce má väčšinou teoretický charakter a druhá polovica empirický charakter. Súčasťou práce je webová stránka a `.m` súbor.

**Kľúčové slová:** E-learning, Page Rank, orientovaný graf, vizualizácia informácií

# Abstract

JÁNOŠOVÁ, Tatiana: *Page Rank.* [bachelor thesis]. Comenius University in Bratislava. Faculty of Mathematics, Physics and Informatics, Department of Alegbra, Geometry and Math Education. Supervisor: doc. RNDr. Andrej Ferko, CSc. Thesis defence: 2018, 35 p.

The main aim of the bachelor thesis is to explain the implementation of the Page Rank algorithm. The primary goal is to present and explain the math model algorithm along with its subsequent inclusion to the topic of E-Learning. The secondary goal is presented by the origin reason clarification and thereby introducing the reader to another alternative of the webpage evaluation. Furthermore, the thesis deals with the visualisation work's progress containing the webpage itself and the additional activation part, created by Page Rank function in Matlab. The first half of the thesis is based on facts having a theoretical character while in the second half of the bachelor thesis the emphasis is put on the empirical part. The webpage and the `.m` file are included, being an inseparable part of the thesis and its topic.

**Keywords:** E-learning, Page Rank, oriented graph, information vizualization

# Obsah

# Zoznam obrázkov

# Úvod

V dnešej dobe, kedy technika a s ňou súvisiaci rozvoj digitalizácie a modernizácie napreduje neuveriteľne rýchlo, pokladáme za dôležité, aby študenti, či iní ľudia poznali princípy a vznik každodenne používaných vecí. Vyhľadávanie pojmov v internetovom prehliadači je zdanlivo jednoduchý proces, no pri hlbšom skúmaní sa ukazuje, že to tak nemusí byť. Algoritmus Page Rank krásne ukazuje, ako sa dá lineárna algebra alebo numerická matematika využiť v praxi.

Hlavným cieľom práce bolo vysvetliť, čo je to Page Rank, ako funguje a tiež ako ho použiť v praxi. Táto téma sa v uplynulých rokoch stala povinnou na mnohých predmetoch na rôznych vysokých školách. Môžme spomenúť FMFI UK, alebo FIIT STU [16]. Pri absolvovaní podobného predmetu s názvom Webovská grafika vysvitlo, že nájsť materiály v slovenčine, kde by bola celá téma zhrnutá a obsahovala aj príklady, môže byť problematické, nakoľko jediným zdrojom informácií bola prezentácia [17]. Táto práca má primárne slúžiť ako podporný materiál na predmet Webovská grafika, ktorý sa vyučuje na FMFI.

Našu bakalársku prácu členíme do troch častí. *Prehľad problematiky* – prvá časť nás oboznamuje so vznikom a začiatkami algoritmu Page Rank, teda má zhrnúť, čo bolo v tejto oblasti vyskúmané v minulosti. Tiež poskytuje alternatívu k Page Ranku a vysvetľuje rozdiely medzi nimi. V ďalšej časti s názvom *Špecifikácia projektu* podrobne rozoberáme celý matematický model, analyzujeme požiadavky a prostriedky pre E-learning a pojednávame o pláne celej konštrukcie projektu, čo zahŕňa webovú stránku aj program v Matlabe. Posledná časť, ktorá má názov *Implementácia* opisuje zavedenie webovej stránky a to v kapitolách: *Dizajn, Rozmiestnenie* a *Obsah*. Súčasťou projektu je aj funkcia Page Rank naprogramovaná v Matlabe, ktorej stavbu a funkčnosť rozoberáme v spomínanej časti.

# Kapitola 1

# Prehľad problematiky

V tejto kapitole opíšeme dôvod vzniku metódy Page Rank (v ďalšom označovaný aj skratkou PR), samotný vznik a tiež jeho vývoj. Uvedieme i niektoré ďalšie alternatívy Page Ranku. Tiež uvádzame, že v celej práci sa opierame o definície a tvrdenia z [2, kap.15].

## 1.1 World Wide Web

World Wide Web alebo len web vytvoril Tim Berners - Lee v ženevskom laboratóriu v roku 1989. Primárnou potrebou bolo vymyslieť spôsob, ako si vedci môžu navzájom poskytovať informácie, a vytvoriť tak vhodnejšie podmienky pre lepšiu spoluprácu. Tento vynález bezprostredne súvisel s vývojom jazyka HTML, v ktorom sa písali webové stránky a tiež HTTP, čo bol súbor pravidiel určený pre počítače na komunikáciu cez internet. Každý dokument, ktorý bol umiestnený na web, mal pridelenú URL adresu, aby sa mohol zobrazovať v prehliadačoch.

World Wide Web je v konečnom dôsledku miesto, kde sa nachádza množstvo webových stránok, popretkávaných hypertextovými odkazmi. Vznikla tak rozsiahla sieť.

## 1.2 Motivácia

Vznik akéhokoľvek ohodnocovania stránok bol nevyhnutnosťou už pár rokov po vzniku World Wide Webu (1989) [1]. Na webe sa nachádzalo obrovské množstvo neutriedených informácií a bolo čoraz ťažšie vybrať si z nich tie potrebné a relevantné. Pre porovnanie, v roku 1998 World Wide Web obsahoval viac ako 150 miliónov webových stránok, v roku 2004 ich počet vzrástol na 10 biliónov [2]. Vyskytla sa preto otázka: Ako nájsť ten najvhodnejší zdroj informácií? Bolo potrebné vyriešiť problém, kde hľadajúci zadá jeden alebo viac textových reťazcov a vyhľadávač nájde stránky obsahujúce zadanie a určí poradie, v ktorom ich ponúkne na výstupe. Neskôr sa dokonca podarilo štandar-

dizovať v norme MPEG-7 [1] aj neindexové vyhľadávanie, pri ktorom sa zadáva obrázok alebo iný typ dát.

Dovtedy používané modely na triedenie informácií na webe boli často nepresné, či menej efektívne. Ako príklady uvedieme booleovský model a pravdepodobnostný model. Prvý zo spomínaných klasifikáciu vykonáva pomocou reťazcov a logických spojok: a, alebo, záporu a ich možných kombinácií. V tomto modeli ale chýba sémantické prepojenie medzi slovami, preto v konečnom dôsledku dostaneme veľmi zredukovaný výstup [3]. Pravdepodobnostný model je o niečo spoľahlivejší, stránky vyhodnocuje pomocou tzv. koeficientu relevantnosti. Spomínaný spôsob je však náročnejší na programovanie, lebo je zložitejší a komplexnejší.

### 1.2.1 Začiatok fenoménu Google

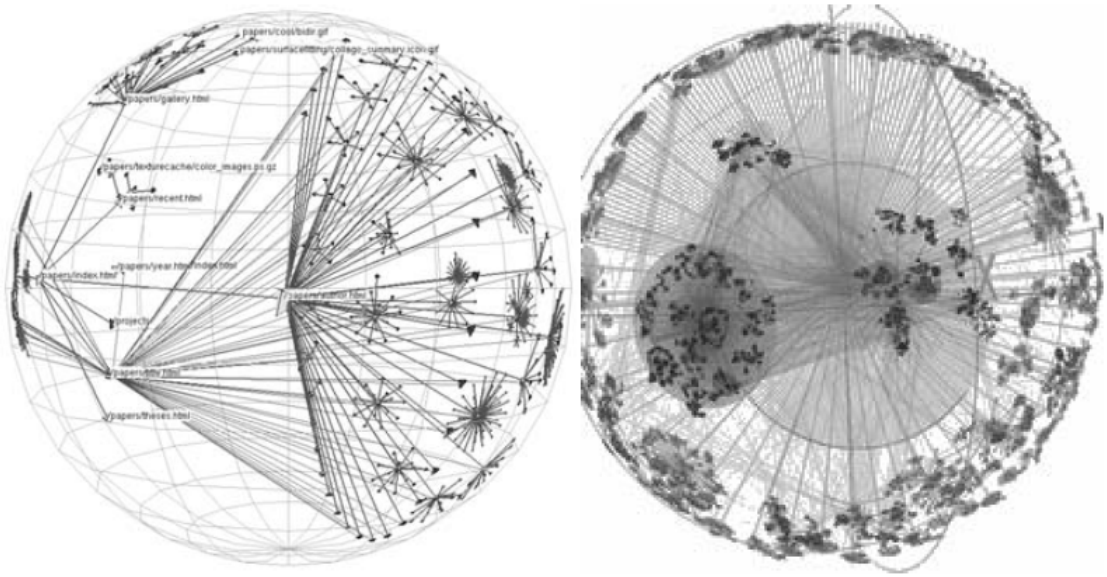
Celému úspechu predchádzalo úsilie vynaložené na dizertačnú prácu Larryho Pagea na Stanfordskej Univerzite. Zaumienil si preskúmať matematické vlastnosti World Wide Webu a pozrieť sa na problém ako na sieť hypertextových prepojení, reprezentovanú grafom, v ktorom sa stránka znázorňuje vrcholom a odkaz orientovanou hranou, pod vedením vedúceho práce Terryho Winograda [5]. Pre tento cieľ sa nadchol aj Sergey Brin, taktiež študent doktorandského štúdia na rovnakej univerzite, a pridal sa k výskumu.

Už počas štúdií na univerzite Sergey Brin a Larry Page založili spoločnosť Google. Tajomstvo mena spočívalo vo vyslovovaní slova googol, ktoré znamená $10^{100}$, teda názov vyjadruje rozsiahlosť vyhľadávacieho prostriedku. V dôsledku narastajúcej práce a popularite spoločnosti, museli obaja študenti zanechať štúdium a naplno sa venovať biznisu. Vyhľadávač sa stal veľmi populárnym a vylepšenia na seba nenechali dlho čakať. Tvorcovia sa rozhodli inovovať spôsob zoraďovania stránok pri vyhľadávaní, čo sa im podarilo a svoj návrh predostreli na siedmej svetovej konferencii World Wide Web v roku 1998 v Austrálii.

## 1.3 Vznik Page Ranku

Riešenie problému s výberom najvhodnejších stránok sa začalo črtať až v roku 1998. Pracoval na ňom aj profesor Cornellovej Univerzity Jon Kleinberg a nezávisle od neho aj tím zo Stanfordovej Univerzity Sergey Brin a Larry Page. Kleinbergov projekt HITS - Hypertext Induced Topics Search bol založený na takmer rovnakom princípe ako Brinov a Pageov projekt s názvom Page Rank. Nová metóda ohodnocovania stránok na základe ich prepojení zožala obrovský úspech a v porovnaní so starými spôsobmi vyhľadávania,

[1] formát, ktorý opisuje obsah multimédií, a uľahčuje tak identifikáciu a kategorizáciu stránok



Obr. 1.1: Munznerova a Hyunova mapa hypertextových prepojení na webe [2, str.27]

založenými na obsahu daných stránok, môžeme povedať, že to bol priam prevratný vynález. Oba algoritmy, teda HITS a Page Rank porovnáme v ďalšej kapitole.

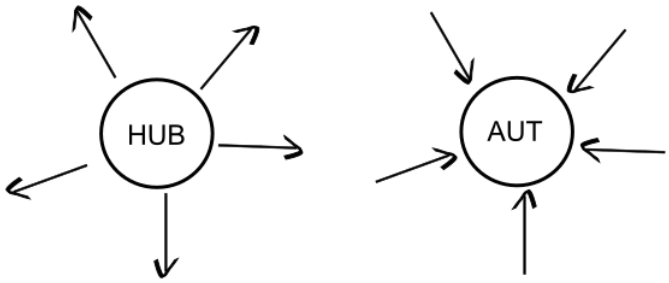
## 1.4 Porovnanie Page Rank a HITS

Obe metódy, Page Rank aj HITS vznikli približne v rovnakom čase, nezávisle od seba. Môžme povedať, že prinášajú rovnaké výsledky, no každá z nich ich dosiahne rôznym spôsobom. V čom teda spočíva rozdiel medzi nimi?

### 1.4.1 Page Rank

Page Rank je algoritmus, ktorý priradí číselnú hodnotu každej stránke z množiny stránok popretkávaných hypertextovými odkazmi, zverejnených vo World Wide Webe. Ako sme už spomínali, metóda využíva vizualizáciu pomocou grafu, ktorý graf je reprezentovaný maticou. Page Rank interpretuje link – odkaz ako hlas. To znamená, že ak existuje odkaz zo stránky $A$ na stránku $B$, tak stránka $A$ dáva hlas stránke $B$. Prirodzene, ak má stránka najväčší počet hlasov, môžme ju považovať za najdôležitejšiu. Čím vyšší je počet hlasov danej stránky, tým vyššie má umiestnenie vo vyhľadávači.

Situáciu trochu komplikuje fakt, že nie každý hlas má rovnakú váhu. Uvedieme príklad: stránka A odkazuje na 6 iných stránok, tak jej hlas má váhu $\frac{1}{6}$. Toto sa zopakuje pri každej stránke. Spočítaním hlasov za jednotlivé stránky získame ohodnotenia. Samozrejme, celý proces je o niečo zložitejší a matematicky ho vysvetlíme v kapitole 2.1.1




Obr. 1.2: Centrum (hub) a autorita (authority)

### 1.4.2 HITS

Nápad Jona Kleinberga, využiť sieť hypertextových odkazov na získanie výsledkov "s najväčšou popularitou", dostal príznačný názov Hypertext Induced Topic Search. Prvýkrát bol objasený verejnosti na deviatom výročnom sympóziu ACM-SIAM v San Franciscu v januári 1998.

HITS rozoznáva dva typy odkazov: tzv. hubs a authorities, v preklade centrá a autority. „Stránku považujeme za centrum (hub), ak podobne ako letisko, obsahuje množstvo liniek vedúcich von. Rovnako opíšeme pojem autorita (authority), stránku nazývame autorita, ak má množstvo liniek smerujúcich dovnútra“ [2, str. 29]. Teda centrom označujeme link, ktorý odkazuje na ďalšie iné linky, podľa obrázka 1.2, šípky smerujú von z centra. Naopak, link je autorita, ak odkazy smerujú k nemu samotnému, čiže iné stránky naň odkazujú. Obrázok (odkaz na obrázok) ukazuje, že šípky idú dovnútra. Pričom platí, že stránka može byť centrom a autoritou súčasne.

Výsledkom algoritmu sú dve ohodnotenia: skóre centra (hub score), v ďalšom značíme H.S. a skóre autority (authority score), ozn. A.S. Stránka má vysoké H.S., ak je spojená s dobrými autoritami a autorita má vysoké A.S., ak na ňu odkazujú stránky, ktoré sú dobrými centrami.

HITS je závislý od dopytu, teda výsledok ovplyvní už zadávaný hľadaný výraz.

#### Algoritmus

Matematicky zapísané, každá stránka $i$ má H.S. $x_i$ a A.S. $y_i$. Ďalej označíme $A$ ako množinu všetkých orientovaných hrán grafu (odkaz na graf prepojených stránok) a $a_{ij}$ bude znázorňovať orientovanú hranu z vrchola $i$ k vrcholu $j$. Teda každá stránka má svoje počiatočné H.S. $x_i^{(0)}$ a A.S. $y_i^{(0)}$. Algoritmus zapísaný pomocou rovníc je nasledovný:

$$x_i^{(k)} = \sum_{j:e_{ij}\in A} y_j^{(k)},$$

$$y_i^{(k)} = \sum_{j:e_{ij}\in A} x_j^{(k-1)},$$

pre $k = 1, 2, 3, ...$

Pre maticový zápis určíme L ako

$$L = \begin{cases} 1, & \text{ak existuje orientovaná hrana z vrcholu } i \text{ do vrcholu } j, \\ 0, & \text{inak.} \end{cases}$$

Matica $L$ pre obrázok 2.1 vyzerá napr. nasledovne:

$$L = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Ak zapíšeme pôvodné rovnice v maticovom zápise, máme

$$x^{(k)} = Ly^{(k)},$$
$$y^{(k)} = L^T x^{(k-1)},$$

pričom $x^{(k)}$ a $y^{(k)}$ sú $(n + 1)$ rozmerné vektory, ktoré aproximujú aktuálny stav H.S. a A.S. v k-tej iterácii.

HITS algoritmus môžme zapísať pomocou niekoľkých krokov:

1. Zvolíme si počiatočný vektor $x^{(0)} = e$, kde $e$ je stĺpcový jednotkový vektor zo samých jednotiek. Avšak môžme použiť aj iný ľubovoľný kladný vektor.

2. $$x^{(k)} = Ly^{(k)},$$
$$y^{(k)} = L^T x^{(k-1)},$$
$$k = k + 1,$$

   kde $x^{(k)}$ a $y^{(k)}$ vždy znormalizujeme. Tento krok opakujeme, až kým $x^{(k)}$ a $y^{(k)}$ skonvergujú.

Je dôležité, že rovnice

$$x^{(k)} = Ly^{(k)},$$
$$y^{(k)} = L^T x^{(k-1)},$$

môžme jednoduchými úpravami previesť na tvar

$$x^{(k)} = LL^T x^{(k-1)},$$
$$y^{(k)} = L^T L y^{(k-1)}.$$

Posledné dve rovnice nám určujú mocninovú metódu s iteráciou, ktorá je veľmi podobná mocninovej metóde Page Ranku. Matica $LL^T$ sa nazýva matica centra a $L^T L$ je matica autority. Vektory $x$ a $y$ sú práve vlastné vektory spomínaných matíc.

### 1.4.3 Page Rank vs. HITS

Myšlienka oboch algoritmov je v podstate rovnaká, pretože Kleinberg ako aj Page a Brin sa pozerajú na web ako na sieť hypertextových odkazov a rozhodli sa pre reprezentáciu pomocou grafov.

Ako sme už ukázali, Page Rank a HITS majú isté odlišnosti, či už ide o myšlienku, realizáciu alebo výstup informácií. Uvedieme niekoľko základných rozdielov týchto metód:

- počet výsledkov – výsledkom Page Rank je jeden vektor, zatiaľ čo HITS vypíše dva výsledky, ako sme spomenuli v predchádzajúcej kapitole.
- závislosť od vstupu – Page Rank nezávisí od vyhľadávaného výrazu. Toto má za následok kratší čas vyhľadávania, pretože pracuje offline [7]. Avšak nedá sa úplne presne rozhodnúť, či je táto vlastnosť výhodou alebo nevýhodou.
- HITS je na rozdiel od Page Rank iteračná metóda.
- použitie – Page Rank používa Google a HITS je algoritmus pre vyhľadávaš Clever spoločnosti IBM.

Každá z metód má svoje výhody aj nevýhody, ale jednoznačne používanejší je Page Rank.

## 1.5 E-learning

E-learning býva definovaný ako forma vzdelávania, ktorá využíva informačné a komunikačné technológie, prebiehajúca prostredníctvom počítačových sietí [9]. „E-learning zahŕňa také výučbové procesy ako: web vzdelávanie, počítačom podporované vzdelávanie, virtuálne triedy a spoluprácu s využitím digitálnych informačných a komunikačných technológií (IKT).Výučba zvyčajne prebieha pomocou internetu, intranetu/extranetu (LAN, WAN), audio alebo videopások, audio alebo video konferencií, satelitného vysielania alebo CD ROM.“ Uvedený citát pochádza z diela M. Mišúta s názvom IKT vo vzdelávaní. Tento model učenia zahŕňa rôzne masmediálne prvky, napr. video, animácia, test, zdieľanie pracovnej plochy či elektronické modely.

E-learning je druh samoštúdia, pretože cieľ edukácie si stanoví sám študent, ktorý túto možnosť využíva. Teda tvorcovia e-vzdelávania veria v samostatnosť a zodpovednosť používateľa, i keď spomínaný model výučby je prístupný už aj mladším používateľom (ZŠ, prípadne predškolská príprava), samozrejme s prispôsobeným a upraveným obsahom a podmienkami používania.

Používatelia tohto typu vzdelávania si sami vedia zvoliť tempo práce a obtiažnosť. Presne si špecifikujú ciele a vyberú materiál, ale aj spôsob učenia, ktorý im najviac

vyhovuje. Nesporne najväčšou výhodou je online dostupnosť, či prispôsobenie časovému harmonogramu edukanta, teda nemá presne stanovený rozvrh, čo, kedy a kde sa vyučuje.

Typy e-learningu:

- e-book,
- blog,
- diskusné fórum,
- e-portfólio,
- hra,
- aplikácia a iné.

Samozrejme v našom zozname nie sú uvedené všetky druhy, pretože je na uvážení tvorcu, aký spôsob uprednostňuje. Proces vzniku e-learningu je predovšetkým kreatívnym procesom.

# Kapitola 2

# Špecifikácia projektu

## 2.1 Princíp Page Rank

V dobe, keď objem dát na internete rástol i stále narastá každou sekundou, bolo potrebné vytvoriť vyhľadávací algoritmus, ktorý by bol efektný, presný a rýchly. Ako sme spomenuli v predchádzajúcej kapitole, túto úlohu sa podarilo splniť, a to modelovaním náhodnej prechádzky po orientovanom grafe. Algoritmus dostal názov Page Rank a v tejto časti podrobne opíšeme princíp a jeho fungovanie.

### 2.1.1 Matematický model PR

Predpokladajme, že stránka má určité citácie, teda existujú odkazy smerujúce na túto stránku. Práve počet týchto citácií nám určuje dôveryhodnosť alebo relevantnosť danej stránky. Preto Page Rank stránky $A$ definujeme nasledovne:

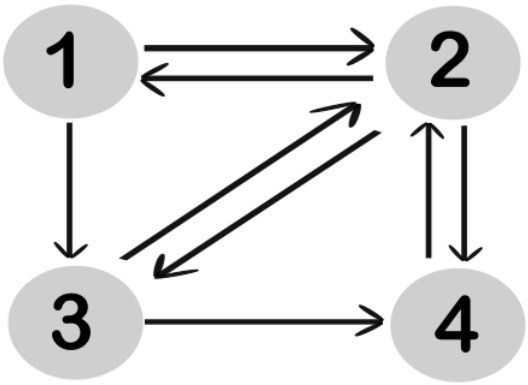
**Definícia 2.1.1.** [8] Predpokladajme, že na stránku $A$ odkazujú citácie $T_1, ..., T_n$. Parameter $d$ je damping faktor, ktorý nadobúda hodnoty medzi 0 a 1. Zvyčajne berieme $d = 0,85$. $C(A)$ je počet odkazov zo stránky $A$ na iné stránky. Potom vypočítame hodnotu $PR(A)$ ako

$$PR(A) = (1 - d) + d(PR(T_1)/C(T_1) + ... + (PR(T_n)/C(T_n)).$$

Page a Brin sa rozhodli, že hypertextové odkazy budú reprezentovať pomocou hrán orientovaných grafov. Vrcholy predstavujú stránky a hrany predstavujú prepojenia medzi nimi. Cieľom vlastne je, aby mal každý vrchol svoje ohodnotenie.

Každému takémuto grafu vieme priradiť maticu $M$

$$M(i,j) = \begin{cases} 1, & \text{ak existuje orientovaná hrana z vrcholu } i \text{ do vrcholu } j, \\ 0, & \text{inak.} \end{cases} \tag{2.1}$$




Obr. 2.1: Graf zobrazujúci prepojenia medzi 4 stránkami.

$$M = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Maticu $M$ sme zostavili na základe obr. 2.1.

Prvé ohodnotenia stránok sa dajú získať už v tomto kroku, stačí iba spočítať čísla v jednotlivých stĺpcoch. Avšak tieto výsledky nie sú zatiaľ korektné. Treba udeliť „váhu“ každému hlasu a to tak, že každý hlas vydelíme počtom odkazov $n_i$ smerujúcich z danej stránky na iné stránky. Teda

$$M(i,j) = \begin{cases} 1/n_i, & \text{ak existuje orientovaná hrana z vrcholu } i \text{ do vrcholu } j, \\ 0, & \text{inak.} \end{cases}$$

Nová matica $M$ pre tento príklad teda bude v tvare

$$M = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Najpresnejší výsledok získame iterovaním, pričom pre hodnotu $PR(A)$ v $k$-tom kroku platí

$$PR_{k+1}(A) = (1-d) + d(PR_k(T_1)/C(T_1) + ... + (PR_k(T_n)/C(T_n)).$$

Iniciálne ohodnotenie pre každú stránku $A_i$ zvolíme $PR_0 = \frac{1}{m}$, kde $m$ predstavuje počet stránok v Google prehliadači na webe. Z toho vyplýva, že každá stránka začína

s rovnakým skóre, čím zaručíme objektivitu. Nazvime tento vektor $\pi^{(0)T} = \frac{1}{m}e^T$, kde $e^T$ je riadkový vektor jednotiek. Potom v k-tej iterácii dostaneme

$$\pi^{(k+1)T} = \pi^{(k)T}M.$$

Pri tomto algoritme ale nastáva viacero problémov. Jedným z nich sú tzv. dangling nodes, stránky, ktoré nikam neodkazujú. V tom prípade vznikajú v matici $M$ nulové riadky. Ďalším problémom je konvergencia. Ako rýchlo a či vôbec tento výpočet skonverguje a či metóda vráti výsledok. Nedostatkom algoritmu sú aj zacyklené odkazy, alebo prepadliská hodnotení (rank sinks), ktoré si v každej iterácii kumulujú vyššie a vyššie skóre. Riešenie ponúkajú Markovove reťazce [1].

Larry Page a Sergey Brin sa rozhodli vyriešiť problémy PR pomocou náhodného surfovania. Je to proces, pri ktorom používateľ náhodne prechádza poprepájané stránky, bez zjavného cieľu. Ukázalo sa, že ak surfujúci strávil na nejakej stránke viac času, po určitej dobe sa znovu preklikal naspäť na túto stránku, preto jej bola pripísaná väčšia dôležitosť. Systém nefungoval práve kvôli spomínaným nikam neodkazujúcim stránkam. V praxi to znamenalo, že potrebovali odstrániť nulové riadky z matice $M$. Toto sa podarilo pomocou stochastických úprav, kedy nulové riadky nahradili vektormi $\frac{1}{m}e^T$ a z matice $M$ vznikla matica $S$

$$S = M + a(\frac{1}{m}e^T),$$

kde $a_i$ je 1, ak stránka $A_i$ neodkazuje nikam a 0 inak.

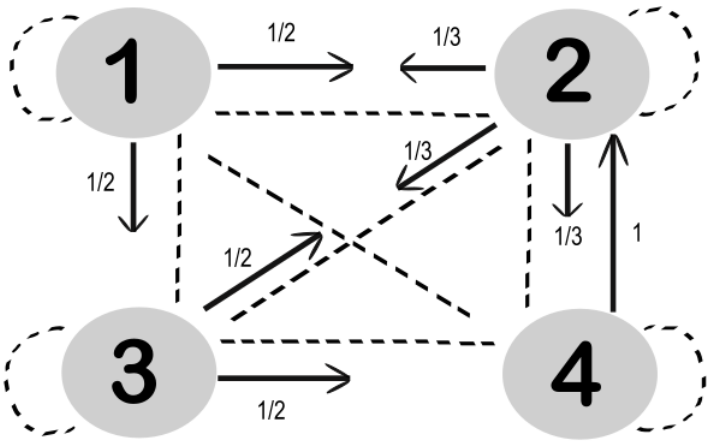
Avšak stochastická úprava stále nezaručuje konvergenciu iterácií. Až úprava na konečnú maticu $G$, nazývanú aj Google matica, ktorá vznikla konvexnou kombináciou matice $S$ a $E$ ju zaručí. Parameter $d$ je z intervalu 0 až 1 vrátane, vyjadruje závislosť medzi časom stráveným na konkrétnej stránke a všetkými ostatnými.

$$G = dS + (1-d)\frac{1}{m}ee^T. \tag{2.2}$$

Pre konkrétnu maticu $M$ (zobrazená vyššie), vypočítame maticu $G$ ako

$$G = 0.85\begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 & 0 \end{pmatrix} + (1-0.85)\frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

[1] vysvetlené v kapitole 15. v [2]




Obr. 2.2: Graf zobrazujúci prepojenia stránok v matici G (čiarkované hrany znázorňujú obojsmerné prepojenia, čísla predstavujú ohodnotenia hrán).

$$G = \begin{pmatrix} 0.0375 & 0.4625 & 0.4625 & 0.0375 \\ 0.320833 & 0.0375 & 0.320833 & 0.320833 \\ 0.0375 & 0.4625 & 0.0375 & 0.4625 \\ 0.0375 & 0.8875 & 0.0375 & 0.0375 \end{pmatrix}$$

Matica $G$ je stochastická, ireducibilná a primitívna, teda $G^k > 0$ pre nejaké $k \in \mathbb{Z}^+$. Z primitívnosti $G$ vyplýva, že existuje jediný nezáporný vlastný vektor $\pi^T$. Vektor hodnotení – Page Rank stránok, získame konvergenciou iterácií tohto vlastného vektora. Teda stačí aplikovať mocninovú metódu na maticu $G$ a konvergencia vektora $\pi^T$ je zabezpečená

$$\pi^{(k+1)T} = \pi^{(k)T} G. \tag{2.3}$$

## 2.2 Projekt E-learning

E-learning na tému Page Rank sme vytvorili prostredníctvom webovej stránky, ktorá bude slúžiť ako podporný študijný materiál na predmet Webovská grafika, ale i všetkým ostatným študentom, ktorí budú mať záujem rozšíriť si vedomosti v danej problematike.

### 2.2.1 Požiadavky

Ešte skôr, ako sme začali rozmýšľať nad prevedením a stavbou e-learningu, bolo nevyhnutné ujasniť si požiadavky pre projekt a následne ich spracovať. Základnou požiadavkou pre vytvorenie e-learningu bude, aby bolo učivo ľahko dostupné a aby si študujúci

našiel celú problematiku na jednom mieste, ideálne v slovenskom jazyku. Dostupnosť by mala byť zapezpečená implementáciou prostredníctvom webovej stránky, teda postačí mať internetové pripojenie a samozrejme chuť učiť sa.

Stránka bude obsahovať definíciu Page Ranku, ktorá bude zapísaná nielen pomocou matematických symbolov, ale aj podrobnejšie vysvetlená, nakoľko voliteľný predmet Webovská grafika navštevujú študenti viacerých programov s rôznymi úrovňami znalostí napr. lineárnej algebry. Súčasťou webovej stránky budú aj konkrétne príklady s riešením a nakresleným grafom, pre ľahšie pochopenie problematiky, prípadne pre ujasnenie si všetkých pojmov. Riešenia budú typovo dvojaké: veľmi obšírne rozpísané, presne podľa postupu opísaného v 2.1.1 a stručnejšie, len na skontrolovanie základných medzikrokov a výsledku.

Ďalej tu bude uverejnený celý algoritmus Page Rank naprogramovaný v Matlabe. Tento program bude prístupný len na stiahnutie, pretože Matlab nie je free software - verejne dostupný. Po stiahnutí m-súboru[2] si bude môcť používateľ vypočítať Page Rank so zadanou maticou M, ktorej prvky si sám zvolí. Pritom si môže pozrieť celý výpočtový proces a spôsob zapísania v Matlabe.

Vzhľad webovej stránky bude prispôsobený na účely edukácie. Teda cieľom je čo najjednoduchšia vizualizácia bez rôznych rušivých elementov, napríklad priveľa farieb či kontrastov, pohyblivých prvkov, alebo zbytočne zložitého ovládania. Uvedené skutočnosti by mohli narúšať proces učenia sa a rozptyľovať študujúceho.

Výsledná webstránka by mala spĺňať požiadavky, konkretizované v predchádzajúcich riadkoch, ktoré boli stanovené na základe potrieb týchto používateľov: študenti, vyučujúci, verejnosť, zaujímajúca sa o túto tému, náhodní surfujúci.

### 2.2.2 Prostriedky

#### Matlab

Hlavným prostriedkom E-learningu bude funkčný program s algoritmom Page Rank tak, ako ho navrhli Page a Brinn. Mal by fungovať v reálnom čase a spracovať akékoľvek vstupné dáta. Mali sme na výber viac možností, v akom jazyku a prostredí spracovať algoritmus, napr. aplikácie podporujúce `C++, C#.` Ale Matlab [10] sa zdal najvhodnejší.

Softvér bol vyvinutý práve pre vedeckú, teda aj matematickú komunitu a preto nebolo potrebné implementovať ďalšie knižnice. Matlab zvláda numerické výpočty podobného charakteru ako Page Rank pomerne rýchlo a samotný kód je prehľadný, ľahko čitateľný a príkazy sú intuitívne, zrozumiteľné aj človeku, ktorý nemá skúsenosti s príkazmi v tomto jazyku. Tvorbu programu a s tým súvisiace zistenia opíšeme v kapitole 3.

[2]Označenie pre súbor (m-file), ktorý obsahuje príkazy z Matlabu.

#### HTML 5

HTML [11] je jazyk bežne používaný na tvorbu webových stránok, ktoré majú byť zobrazené vo webovom prehliadači. Nová verzia `HTML 5` je len inováciou klasického `HTML` s pridaním niektorých multimediálnych prvkov a podporou pre offline aplikácie. `HTML` umožňuje vytvoreť text, multimediálny obsah, hypertextové odkazy a ďalšie prvky. Po zobrazení na webe vidíme len samotné vložené prvky, bez grafických úprav a štruktúr.

#### CSS

Rozšírenie jazyka `HTML`, nazývané aj kaskádové štýly, dodávajú vizuálnu formu `HTML` súboru. `CSS` [12] formátuje vložený text, či iné súčasti dokumentu. Nachádza sa v oddelenom súbore, preto je výsledný kód stránky prehľadnejší. Nevýhodou používania kaskádových štýlov je, že takto naprogramovaná stránka sa nemusí zobraziť rovnako v rôznych webových prehliadačoch. Pomocou využitia `CSS` spolu s `HTML 5` naprogramujeme webstránku.

#### Inkscape

Inkscape [13] je vektorový grafický editor, čo znamená, že produkuje výsledné súbory vo formáte `.svg`. Vo vektorovom obrázku v porovnaní s rastrovým obrázkom sa potom pri ďalších úpravách ako sú napríklad škálovanie alebo rotácia, zachovávajú pomery strán a uhly. Toto je veľmi šikovné najmä pre vkladanie obrázkov na webovú stránku a pre ďalšie spracovanie obrázkov. Program Inkscape bude užitočný pri vytváraní ukážkových príkladov grafov, ktoré budú reprezentovať nejakú sieť.

#### MathML

Programovací jazyk, pomocou ktorého sa dá integrovať matematické značenie na webové stránky. `MathML` (Mathematical Markup Language) [14] bol vytvorený ako aplikácia `XML` a je súčasťou `HTML 5`. Spomínaný jazyk sme vybrali kvôli jeho použiteľnosti pri písaní rovníc, matíc a pododne. Druhou alternatívou pre implementáciu matematických výrazov na stránku bolo vsadenie pomocou rastrových obrázkov. Avšak v tomto prípade by mohol nastať problém s načítaním obrázkov a tomuto sme sa rozhodli predísť.

### 2.2.3 Konštrukcia webstránky

V kapitole 2.2.2 sme vymenovali a zdôvodnili použitie konkrétnych prostriedkov, pomocou ktorých budeme vytvárať webovú stránku. V nasledujúcich riadkoch špecifikujeme vonkajšiu aj vnútornú formu E-learingového portálu, teda obsah a vzhľad.

Čo sa týka vizuálnej stránky E-learningu, dôležité je, aby nebol študujúci rozptyľovaný. Preto uprednostníme jednoduchosť a prehľadnosť. Farby na stránke nebudú krikľavé a použijeme maximálne tri farby písma. Pozadie bude neutrálne, jemné a celkový dojem by mal byť príjemný. Taktiež typ písma použijeme len jeden a bude úplne obyčajný, aby sa ľahko čítal.

Na webstránke sa bude nachádzať navigácia v podobe menu a samotný text problematiky. Menu bude pozostávať z hlavnej stránky a troch ďalších podstránok: stručné zhrnutie témy, vysvetlený algoritmus a príklady. Text bude vybraný z tejto práce, taktiež aj vzorce. Príklady budú doplnené o obrázky grafov vytvorených v Inkscape vo formáte `.svg` a súčasťou podstránky s príkladmi bude aj `.m` súbor na stiahnutie s matlabovskou funkciou. Všetky súčasti webstránky budú mať jednotný dizajn a štruktúru.

### 2.2.4 Program PR

Voľbu programu Matlab, v ktorom sme sa rozhodli implementovať algoritmus PR, sme zdôvodnili v podkapitole 2.2.2. Avšak rovnako dôležité rozhodnutie bolo, čo bude tento kód obsahovať, aké budú inicializačné údaje a čo bude na výstupe tejto funkcie.

Pre účely e-learningu upravíme matlabovský kód zverejnený v [2, str. 43]. Túto voľbu ovplyvnilo viacero faktorov. Zmienený algoritmus obsahuje všetky potrebné informácie a proces je veľmi podobný výpočtu, ktorý sme opísali v kap.2.1.1. Algoritmus budeme interpretovať pomocou funkcie s názvom Page Rank, pre lepšiu prácu s príkazmi a celkovou prehľadnosťou. Vstupným údajom bude zadaná matica $M$, začiatočný vektor $\pi$ a hodnota, ktorá bude reprezentovať toleranciu konvergencie vektora $\pi$ v mocninovej metóde – odchýlka [3]. Vstupom by mohol byť aj graf, či priamo hypertextový odkaz na stránku, ale zvolili sme časovo a na výpočet menej náročnú alternatívu, ktorá na dané účely úplne postačuje.

Ďalším kritériom pri výbere najvhodnejšieho algoritmu bola jeho dĺžka. Tri riadky s príkazmi edukantovi nepomôžu, pretože taký zjednodušený kód nebude obsahovať všetky podstatné informácie a nebude jasné, o čo v ňom ide. Naopak, príliš dlhý kód bude chaotický. Na základe týchto argumentov sme vybrali vhodný kompromis.

Podstatná informácia je aj samotné zloženie programu. Funkcie, ktoré obsahuje a jednotlivé metódy, ktorými sa algoritmy odlišujú – každý používateľ by si ich modifikoval podľa svojej potreby.

[3] Označenie zachováme rovnaké ako v podkapitole 2.1.1.

# Kapitola 3

# Implementácia

V tejto časti práce opíšeme, ako sa nám podarilo implementovať algoritmus Page Rank do Matlabu a edukačné texty na webovú stránku, ktorá bude slúžiť ako E-learning. Zhodnotíme tiež to, ako sa nám podarilo splniť všetky kladené požiadavky.

## 3.1 Webová stránka

Webová stránka má slúžiť ako edukačný materiál na tému Page Rank. Cieľom je, aby tu bola daná téma zrozumiteľne vysvetlená na príkladoch, aj prostredníctvom učebného textu.

Hlavnými nástrojmi pre tvorbu stránky boli programovacie jazyky `HTML 5, CSS` a `MathML`. Samotný kód sme písali v prostredí `Komodo` [15] pomocou jazyka `HTML 5`. Takýmto spôsobom sme napísali najskôr len textové odstavce a následne sme kódovali matematické výrazy prostredníctvom `MathML`.

Stránku sme naformátovali pomocou gridov v `CSS`, čo sú vlastne akési kontajnery alebo bunky. Podstatou je, že celú plochu rozdelíme do buniek, aby sme mohli určiť rozloženie jednotlivých prvkov dokumentu (hlavička, navigácia, nezávislé články a iné). Potom už len určíme polohu každej bunky a tiež stanovíme, ktorá časť dokumentu sa v nej bude nachádzať. Rozloženie webovej stránky sa dá naformátovať aj pomocou tabuliek, či frameworkov. V porovnaní s týmito metódami tvorby layoutu [1], použitie gridov má ľahko čitateľný kód a celý proces je omnoho intuitívnejší.

### 3.1.1 Dizajn

Vizuálna rovina webstránky bola vytvorená na základe požiadaviek, ktoré sme si stanovili. Celý dizajn bol konštruovaný s cieľom, aby stránka bola prehľadná a jednoduchá.

Font písma sme zvolili `Arial`, pretože je ľahko čitateľný. Patrí medzi bezpätkové

[1] layout = rozloženie stránky

druhy písma, a teda je graficky jednoduchší. V texte kombinujeme normálny rez písma s tučným a kurzívou, preto nebolo potrebné použiť ďalší typ písma.

Podobný princíp ako pri písme sme dodržiavali aj pri výbere farieb. Celkový vzhľad je úplne klasický – čierne písmo na bielom pozadí. Túto monotónnu kombináciu sme oživili farebnými nadpismi a farebným menu. Nadpisy sa vyskytujú v dvoch farbách a veľkostiach, podľa dôležitosti. Názvy podstránok sú väčším písmom s tmavoolivovou farbou a názvy menších úsekov sú tmavoružovou farbou. Tmavoolivová farba je použitá aj na názvy v menu, aby sme zdôraznili súvis medzi týmito názvami v navigácii a názvami podstránok.

### 3.1.2 Rozloženie

Pomocou gridov sme rozdelili dokument na 5∗10 buniek, zostavených z hlavičky(header), navigácie (nav), nezávislých článkov (article) a päty (footer). Smer textu je zhora nadol, nazýva sa stĺpcový (column), so zarovnaním na stred.

Horná časť pozostáva z hlavičky, ktorá obsahuje veľký nadpis zobrazujúci názov a cieľ projektu. Neobsahuje nič navyše, zámer je úplne jednoznačný: informovať surfujúcich, kam práve prišli. Tí, ktorí stránku vyhľadali cielene, na nej zostanú, a tí, ktorí sem trafili náhodou, hneď vedia, čo majú čakať.

Pod hlavičkou sa začína časť, kde sa nachádzajú články a na ľavej strane navigácia. Navigáciu tvorí štvorriadkové menu. Od textu je oddelené vertikálnou medzerou (okrajom) a zvýraznené tmavoolivovou farbou. Keďže ide o odkazy, presmerujúce používateľa inam, názvy podstránok sú podčiarknuté. Menu je tu zafixované, čo znamená, že ak sa používateľ posúva smerom dole po stránke, menu sa nehýbe a zostáva staticky umiestnené v strede ľavého kraja. Toto riešenie je veľmi výhodné, pretože sa netreba dostať naspäť na začiatok stránky smerom hore, ale študujúci sa vie okamžite presmerovať na inú položku v zozname navigácie.

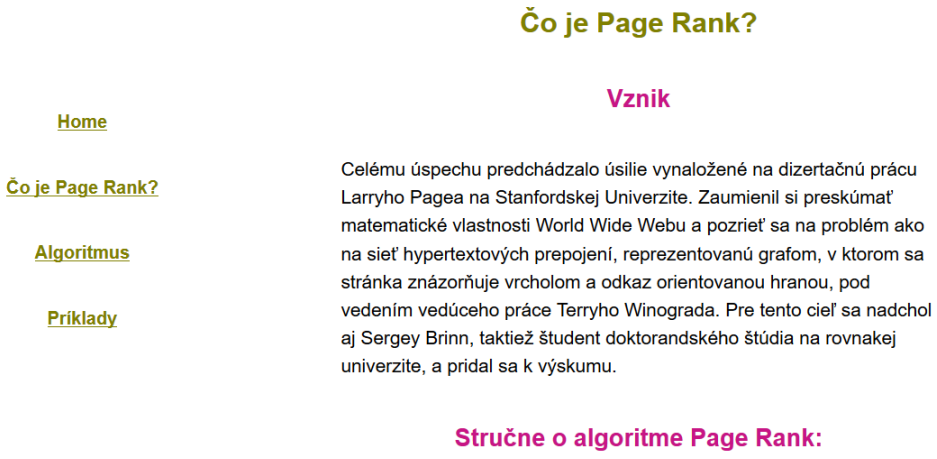
Články tvoria spolu 14 buniek. Je v nich uložený hlavný obsah, napríklad opísaný algoritmus, vysvetlenie problému, či príklady. Významovo odlišné časti sú od seba oddelené farebnými nadpismi s väčším typom písma.

#### CSS súbor

Dizajn stránky je zapísaný v súbore `style1.css`. Najprv sme si určili rozloženie stránky pomocou gridov. Toto rozmiestenie sa nazýva template. Tým sme následne určili ďalšie špecifické vlastnosti: veľkosť, farba, umiestnenie v template... Ďalšie špecifikácie určujeme cez flexboxy (kontajnery v $1D$), patria sem smer, tok, zalamovanie a zarovnanie textu. Súbor je rovnaký pre všetky podstránky, aby boli jednotné a napísané štýlovo rovnako.

```css
body
{
    display: grid;


    grid-template-columns: 10% 10% 10% 10% 10% 10% 10% 10% 10% 10%;

    grid-template-rows: 220px auto auto auto auto;

    font-family: arial;
    font-size: 1.3em;
    line-height: 1.5em;

    margin: 0 0 0 0;
}
```

Obr. 3.1: Fragment kódu, ktorý určuje veľkosť a rozloženie buniek, ďalej typ písma a nastavenie okrajov.




Obr. 3.2: Dizajn stránky zobrazujúci menu a článok

### 3.1.3 Obsah

Webová stránka pozostáva zo štyroch ucelených podstránok. Obsah daných podstránok si opíšeme v tejto podkapitole.

#### Home

Ako napovedá samotný názov, táto podstránka je úvodnou, mala by teda surfujúceho pripraviť na to, čo tu nájde. Okrem pevného zloženia pozostávajúceho z menu a hlavičky, sa tu nachádza kratší článok s názvom motivácia. Bol vybraný z tejto práce z kapitoly s rovnomenným názvom a cieľom tohoto textu je naozaj podnietiť šudujúceho do ďalšieho prehľadávania stránky. Opisuje, čo bolo podnetom pre vznik algoritmu Page Rank a aká bola situácia predtým, než ho Larry Page a Sergey Brin vymysleli.

### Čo je Page Rank?

Sekcia s názvom Čo je Page Rank? zachytáva pôvod algoritmu a to, ako vznikol. Aj preto je hneď pod tmavoolivovým názvom podstránky ružový nadpis Vznik. Potom formou zoznamu študujúcemu stručne vysvetlíme, čo to Page Rank je. Výsledné zhrnutie algoritmu sa nachádza v dolnej časti dokumentu a je oddelené nadpisom Záver.

### Algoritmus

Jednou z požiadaviek bolo, aby E-learning dostatočne podrobne vysvetľoval danú problematiku. Pre tento účel slúži podstránka s názvom Algoritmus. Sú tu vybrané pasáže z práce, avšak bolo potrebné výklad skrátiť. Webové stránky by vo všeobecnosti nemali obsahovať príliš veľa textu, aby to návšetvníkov stránky neodradilo. Týmto pravidlom sme sa riadili a ozrejmenie algoritmu sme skrátili, ale nie príliš, aby sa nestratilo jadro problému.

Na písanie niektorých syntakticky náročnejších vzorcov sme použili matematický programovací jazyk `MathML`. Aby text nebol jednotvárny, rozdelili sme ho nasledujúcimi nadpismi: Problém?, Riešenie. Z toho vyplýva, že počas vysvetľovania sme narazili na nejaký problém, ale zároveň bolo poskytnuté aj riešenie.

### Príklady

Pre praktickú ukážku Page Ranku sme naprogramovali podstránku Príklady. Rozhodli sme sa, že tu budú tri vyriešené príklady, ktorých zadanie bude graf a ďalej budeme postupovať podľa návodu na podstránke Algoritmus. Grafy boli vytvorené v programe `Inkscape` a sú vo vektorovom formáte. Na konci dokumentu je priložený súbor na stiahnutie s funkciou Page Rank naprogramovanou v Matlabe.

Prvý typ príkladu zobrazuje prepojenia medzi tromi stránkami. Tento príklad bude podrobne vyriešený a môžeme ho nazvať ako modelový príklad.

Ďalšia úloha už obsahuje štyri stránky a hypertextové prepojenia medzi nimi. V tomto prípade z postupu vynecháme niektoré kroky, nakoľko by bol totožný s tým predchádzajúcim. Pod príklad uverejníme výsledok na kontrolu a ak bude mať študujúci problém s riešením, môže si pozrieť jednotlivé medzivýsledky v priloženom `.m` súbore.

Posledný príklad sa bude najviac podobať reálnej situácii. Graf je zložený z troch prepojených stránok, ale jedna z nich neodkazuje nikam, je to tzv. dangling node. Riešenie bude preto komplikovanejšie. Aj z tohoto dôvodu sme sem spísali kompletný postup s výsledkom.

## 3.2 Funkcia PR

Ako základnú konštrukciu programu sme zvolili kód uverejnený v [2, str. 43]. Naproti originálu, sme ale museli uskutočniť pár zmien, aby program plne vyhovoval našim požiadavkám, ktoré sme si stanovili pre tento účel.

Algoritmus Page Rank sme naprogramovali prostredníctvom funkcie s výstižným názvom – PageRank. Toto riešenie sa nám zdalo elegantnejšie a prehľadnejšie, v porovnaní so štruktúrou kódu naprogramovaného bez funkcie a teda aj nejakej formy.

### 3.2.1 Inicializačné údaje

Vstup sme zvolili iný, ako pri pôvodnom programe z naštudovanej literatúry – v ďalšom označovaný ako pôvodný. Dôvody týchto zmien sme už spomenuli v podkapitole 2.2.4 s názvom Program PR. V tejto časti práce chceme vykonané zmeny bližšie špecifikovať.

V pôvodnom programe je nami vybraná funkcia akýmsi pokračovaním inej, zložitejšej, ktorá transformuje hypertextové odkazy na graf a neskôr na maticu. Vstupným údajom je teda link. V našom e-learningu sme sa rozhodli zobrať iba druhú časť programu, teda tú, ktorá neobsahuje toto spracovanie prepojení medzi konkrétnymi stránkami. Ako vstup sme si vybrali maticu $M$, reprezentujúcu orientovaný graf odkazov medzi stránkami, ďalej počiatočný vektor $\pi^0$ a hodnotu s názvom odchýlka, ktorá znázorňuje toleranciu konvergencie.

S výberom vstupných parametrov súvisia nasledujúce podmienky:

- Matica $M$ musí byť štvorcová a z toho, ako sme ju definovali, vyplýva, že jej prvkami sú len 1 alebo 0.
- Rozmery začiatočného vektora $\pi^0$ musia súhlasiť s rozmermi matice $M$. Teda ak má $M$ rozmery $n \times n$, tak $\pi^0$ je riadkový vektor obsahujúci $n$ prvkov.
- Tolerancia konvergencie je skalár, čiže musí byť zadaná len číslom.

Konkrétne inicializačné údaje by mali poskytovať dosť veľkú mieru variability príkladov pre používateľov a zároveň algoritmus neobsahuje príliš veľa vstupných dát.

### 3.2.2 Telo funkcie

Funkciu sme rozčlenili do štyroch častí, podľa typu výpočtov či procesov, ktoré v danom úseku prebiehajú.

Prvým krokom pri výpočte Page Ranku v našom algoritme bolo nastavenie hodnôt, ktoré budeme ďalej potrebovať, či inicializácia pomocných prvkov. Jedným z nich je $d$,

tzv. dumping faktor. Priradili sme mu nemennú hodnotu 0.85. V literatúre sa v súvislosti s týmto parametrom spája viacero čísel, avšak my sme zvolili hodnotu práve z diela [8], ktoré napísali Page a Brin. Spomínaný dumping faktor sme takto vyčíslili aj v definícii 2.1.1. Ďalej sme vektor $e$ naplnili jednotkami a určili rozmery vektora: $(n \times 1)$, ide o stĺpcový vektor s $n$ prvkami. Počiatočnú hodnotu pomocnej premennej $k$ pre `while` cyklus sme nastavili na 1.

Overenie podmienok pre vstupné parametre bolo ďalšou časťou funkcie. Pri zadaní nesprávnych inicializačných údajov vypíše program chybovú hlášku a upozorní používateľa na nesprávne zadaný vstup. Overovali sme veľkosť matice $M$, ktorá musí byť štvorcová a musí obsahovať len 0 alebo 1. Tiež bolo nevyhnutné zistiť veľkosť vektora $\pi^0$ v súvislosti s rozmermi iniciačnej matice.

Treťou časťou je séria výpočtov, ktorými sa snažíme dostať k výslednej matici $G$. Predelením každého prvku súčtom prvkov v riadku, v ktorom sa nachádza, udelíme váhu každému hlasu. Toto delenie prebieha po prvkov, na rozdiel od delenia matíc v algebraickom zmysle, aby bol požadovaný výsledok korektný. Vznikne takto matica $M1$.

Na zisťovanie, či je stránka $A_i$ dangling node, teda nikam neodkazujúca stránka, sme použili riadky z pôvodného algoritmu. Pointou je nájsť nulové riadky v transponovanej matici $M1$, ktoré sú vlastne nulovými stĺpcami v danej matici. Následne sme do vektora $a$ príkazom `sparse` – transformuje maticu na riedku tak, že si zapamätá len pozície a hodnoty nenulových prvkov – priradili údaje z matice vyskladanej nami určenými prvkami. Dôležité pri tomto kroku bolo, aby nám v počítaní s maticami sedeli všetky rozmery, inak by program nefungoval.

Poslednou časťou funkcie PageRank je mocninová metóda, pri ktorej iterovaním získame výsledný vektor ohodnotení $\pi$. Najprv si načítame do vektora $\pi$ začiatočné hodnoty vektora $\pi^0$. Samotný výpočet prebieha v prostredí `while` cyklu za podmienky, že tolerancia konvergencie neprekračuje nami stanovenú hodnotu $k$. Ak podmienka platí, aplikuje sa mocninová metóda na vektor $\pi$. Súčasťou metódy je aj normovanie tohoto vektora, aby sme zaručili jeho ohraničenie a zároveň aj zbehnutie algoritmu do konca.

Prepis funkcie PageRank v Matlabe. Zmeny oproti pôvodnému kódu sme vyznačili tučným písmom. Za zmenu nepovažujeme preoznačenie, či preklad názvov vektorov/konštánt v algoritme.

**function [ pi ] = PageRank( M, pi0, odchylka )**
**pocriad = 0;**
**pocstl = 0;**
**d = 0.85;**
**[ pocriad, pocstl ] = size (M);**

```
while pocriad > pocstl || pocriad < pocstl
disp ( 'zadana nespravna matica' )
end

B = (M == 0) | (M == 1);
for i = 1 : pocriad
for j = 1 : pocriad
if B(i, j) > 1 || B(i, j) < 1
disp ( 'zadana nespravna matica' )
end
end
end

pisize = size(pi0, 2);
if pisize  = pocriad
disp( 'zadany nespravny zaciatocny vektor pi0' )
end

s = sum (M, 2);
i = find(  s);
M1 = M./ s;
idlzka = lenght(i);
for j = 1 : pocriad
for k = 1 : idlzka
if j == i(k)
M1( j, :) = zeros([1, pocriad]);
end
end
e = ones([pocriad, 1]);
k = 1;

sumriad = ones(1, pocriad)* M1';
nenuloveriadky = find(sumriad);
nuloveriadky = setdiff(1 : pocriad, nenuloveriadky);
l = length(nuloveriadky);
a = sparse (nuloveriadky, ones(l, 1), ones(l, 1), pocriad, 1);

pi = pi0;
while (k >= odchylka)
```

```
predpi = pi;
pi = d * pi * M1 + (d*(pi * a) + 1-d)*((1 / pocriad) * ones(1, pocriad));
k = norm( pi - predpi, 1);
end
```

Vysvetlivky k označeniu (ostatné označenia sú definované priamo v kóde):
M – vstupná matica s rozmerom $n * n$
pi0 – začiatočný vektor s rozmerom $1 * n$
odchylka – tolerancia konvergencie (skalár)
pi – výsledný vektor ohodnotení s rozmerom $1 * n$
d – dumping faktor (skalár)

### 3.2.3 Výstupné údaje

V pôvodnom programe z [2] je na výstupe viacero údajov:

1. Page Rank vektor – výsledné ohodnotenia stránok,
2. čas, za ktorý bol PR vektor vypočítaný,
3. počet iterácií, než PR vektor skonvergoval.

Naproti tomuto, upravená funkcia PageRank obsahuje len jednu výstupnú informáciu a to vektor ohodnotení. Uviesť viac údajov sme považovali za nepotrebné pre používateľov e-learningu, nakoľko to nebolo cieľom práce.

# Záver

Cieľom práce bolo zosumarizovať poznatky o problematike a tým poskytnúť vhodný študijný materiál, ktorý bude v praxi podporovať výučbu danej témy. Chceme spopularizovať algoritmus všetkým, ktorí budú mať záujem rozšíriť si vedomosti v tejto oblasti.

Predloženou prácou sme sa pokúsili predostrieť v slovenčine dostatočné množstvo informácií o potrebách vzniku a princípe fungovania Page Ranku. Tieto časti sme podrobne rozpracovali a vysvetlili, nakoľko si myslíme, že Page Rank ako taký, sa pomaly zaraďuje medzi všeobecné poznatky, ktoré by mal ovládať každý, kto sa orientuje v oblasti informačných technológií.

Základnou úlohou bolo vytvorenie webovej stránky, na ktorej by boli zverejnené všetky poznatky vyplývajúce z tejto práce. Teoretickú časť sme doplnili príkladmi, pre lepšie a hlbšie spoznanie problematiky. Podstatnou súčasťou praktickej implementácie je aj program v Matlabe, konkrétne funkcia s názvom Page Rank. Stránka sa bude používať pri výučbe a bude zverejnená na webe Katedry algebry, geometrie a didaktiky matematiky: *http://flurry.dg.fmph.uniba.sk/web_studenti/janosova/home.html.*

Pri verifikácii výsledkov a po zapojení `HTML` kódu a celej stránky na web sa objavil problém so správnym zobrazovaním matematických vzorcov v rôznych prehliadačoch, nakoľko jazyk `MathML` má plnú podporu len v prehliadači Mozilla. Pri spustení pomocou iných prehliadačov dochádza, z hľadiska presnosti, k nevyhovujúcemu zobrazeniu, a to môže viesť k nesprávnej interpretácii vzorcov. Keďže primárnym cieľom stránky má byť edukácia, rozhodli sme sa konvertovať vzorce napísané v `MathML` do iného jazyka, ktorý bude podporovaný všetkými prehliadačmi a do zdrojového súboru s názvom `priklady.html` sme vložili ešte nasledujúci riadok: `<script type="text/javascript "src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.2/MathJax.js?config=MML_HTMLorMML»</script>`. Samozrejme, vymazaním spomínaného scriptu sa dá vrátiť k „akademicky čistej“ verzii stránky s `MathML`.

Keďže téma je veľmi obšírna, naša bakalárska práca obsahuje len základné poznatky. Ďalšie skúmanie správania algoritmu, portovanie funkcie napr. do mobilnej aplikácie, možné problémy s rýchlosťou konvergencie, či špeciálne typy orientovaných grafov by boli predmetom oveľa rozsiahlejšej práce.

# Literatúra


[1] Gillies, J. – Cailliau, R. 2000. *How the Web Was Born: The Story of the World Wide Web.* Oxford University Press.

[2] Langville, A. N. – Meyer, C. D. 2017. Google's PageRank and Beyond: The Science of Search Engine Rankings, [online] Dostupné na internete: http://geza.kzoo.edu/ erdi/patent/langvillebook, 16. decembra 2017

[3] Salton, G. et al. 1983. *Extended Boolean Information Retrieval.* Communications of the ACM.

[4] Manning, CH. D. et al. 2017. An Introduction to Information Retrieval, [online] Dostupné na internete: https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf, 18. decembra 2017

[5] Fisher, L. M. 2017. Siri, who is Terry Winograd?, [online] Dostupné na internete: https://www.strategy-business.com/article/Siri-Who-Is-Terry-Winograd?gko=e5f07, 24. januára 2018

[6] Dubey, H. et al. 2018. An Improved Page Rank Algorithm based on Optimized Normalization Technique, [online] Dostupné na internete: http://www.ijcsit.com/docs/Volume 2/vol2issue5/ijcsit2011020570.pdf, 1. februára 2018

[7] Devi, P. – Gupta, A. – Dixit, A. 2014. *Comparative Study of HITS and PageRank Link based Ranking Algorithms.* International Journal of Advanced Research in Computer and Communication Engineering.

[8] Page, L. – Brin, S. 1998. *The Anatomy of a Large-Scale Hypertextual Web Search Engine.* Proceedings of the Seventh International Web Conference (WWW 98).

[9] Mišút, M. 2013. *IKT vo vzdelávaní.* Trnava: Pedagogická fakulta Trnavskej univerzity v Trnave. 90 s. ISBN 978-80-8082-695-6

[10] „MATLAB," [online] Dostupné na internete: https://www.mathworks.com, 9. mája 2018

[11] „HTML,“ [online] Dostupné na internete: https://www.w3schools.com/html/default.asp, 9. mája 2018

[12] „CSS,“ [online] Dostupné na internete: https://www.w3schools.com/css/default.asp, 9. mája 2018

[13] „Inkscape,“ [online] Dostupné na internete: https://inkscape.org/en/, 9. mája 2018

[14] „MathML,“ [online] Dostupné na internete: https://www.w3.org/Math /whatIs-MathML.html, 9. mája 2018

[15] „Komodo,“ [online] Dostupné na internete: https://www.activestate.com/komodo-ide/downloads/edit, 9. mája 2018

[16] Návrat, P. et al. 2014. *Weboveda: východiská, predmet, metódy.* Bratislava: Slovenská technická univerzita v Bratislave. 144 s. ISBN 978-80-227-4264-1

[17] Katreniaková, J. 2010. PageRank Algoritmus. [online] Dostupné na internete: http://www.sccg.sk/ ferko/DrKatreniakovaPageRank.pdf, 9. mája 2018

# Zoznam príloh

CD obsahujúce zdrojový kód stránky a fukciu PageRank v `.m` súbore