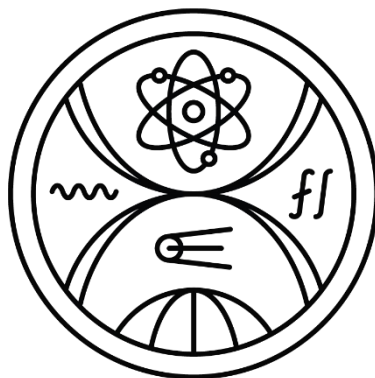
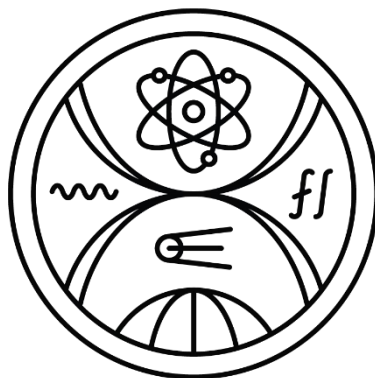


UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



TOKEN GRAFY
Bakalárska práca

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



TOKEN GRAFY
Bakalárska práca

Študijný program: Aplikovaná informatika

Študijný odbor: Informatika

Školiace pracovisko: Katedra aplikovanej informatiky

Školiteľ: Mgr. Dominika Mihálová

Bratislava, 2024

Timotea Chalupová



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Timotea Chalupová
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Token grafy
Token graphs

Anotácia: Cieľom bakalárskej práce je implementovať algoritmy na token grafoch. Súčasťou práce je naštudovať a vytvoriť prehľad vlastností token grafov.

Vedúci: Mgr. Dominika Mihálová
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.
Dátum zadania: 04.10.2023

Dátum schválenia: 05.10.2023

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

ČESTNÉ PREHLÁSENIE

Čestne prehlasujem, že bakalársku prácu som vypracovala samostatne, len s použitím uvedenej literatúry a za pomoci konzultácií mojej školiteľky.

Bratislava, 2024

.....
Timotea Chalupová

POĎAKOVANIE

...

abstrakt

abstract

Obsah

Úvod.....	1
1. Východisková kapitola.....	2
1.1. Základné pojmy.....	2
1.2. Token grafy	4
1.3. Analýza technológií.....	4
1.3.1. Python technológie.....	4
1.3.2. Java technológie.....	5
2. Špecifikácia.....	6
3. Návrh.....	7
4. Implementácia.....	8
5. Testovanie.....	9
6. Použitá literatúra	10

ÚVOD

V dnešnom rýchlo vyvíjajúcom sa svete, plnom rôznych informačných technológií, je dôležité hľadať nové algoritmy a dátové štruktúry, ktoré môžu nájsť uplatnenie nielen v teoretickej informatike ale aj v praxi. V matematike, v informatike a rovnako aj v reálnom svete sa veľké množstvo problémov dá znázorniť pohybom objektov po vrcholoch grafu. Z toho dôvodu sú token grafy významnou matematickou štruktúrou, ktorá nachádza využitie v analýze grafov, grafovej teórii a distribuovaných systémoch. Ich výskum a analýza môžu poskytnúť užitočné poznatky pre optimalizáciu algoritmov.

...

V prvej kapitole si objasníme základné pojmy z teórie grafov, ktoré sú nevyhnutné pre porozumenie danej problematike. (Spomenieme termíny ako sú ...). Taktiež sa pozrieme na porovnanie technológií

V druhej kapitole si priblížime

V tretej....

Cieľom je...

1. VÝCHODISKOVÁ KAPITOLA

1.1. Základné pojmy

V tejto kapitole vysvetlíme základné pojmy a definície, ktoré sú nevyhnutné pre vypracovanie práce.

Regular

Node connectivity

Edge connectivity

Coloring

Girth connectivity

Eulerian and Hamiltonian

Isomorphism

Tree

Planar

Shortest path

(Uvidím čo bude treba, niečo z týchto)

(Graf, Vrchol, Hrana, Orientovaný, Neorientovaný, Multigraf, Acyklický graf, Stupen vrcholu, Stupen uzla, Sled v grafe, Tah v grafe, Cesta v grafe, Hamiltonovská cesta v grafe, Eulerov sled, Eulerov tah, Súvislý graf, Nesúvislý graf, Úplný graf, Kostra grafu, Vstupný vrchol, Výstupný vrchol, Kružnica, Cyklus, Strom, Sieť, sieťový graf, Úloha cinskeho postara, Úloha obchodného cestujúceho, Úloha optimálneho umiestnenie depa, Úloha určenia minimalnej cesty, Úloha určenia minimalnej / maximalnej kostry grafu, Úloha určenia maximalného toku v sieti)

(Hub, Cesta, Dĺžka cesty, Slučka, Priemerná dĺžka cesty ?, Hustota, Podgraf, Súvislý/nesúvislý, Priemer)

(Simple graf, Regular graf, Paths and cycles, Subgraf and spanning subgraf, Strom, Spanning tree, Graph isomorphism, Graph symmetries, Graph automorphism)

1.2. Token grafy

1.3. Analýza technológií

V tejto podkapitole sa zameriame na niekoľko rôznych programovacích jazykov a knižníc, ktoré sme vzájomne porovnávali, aby sme našli najvhodnejšie technológie na implementáciu token grafov.

1.3.1. *Python technológie*

Python je vysokoúrovňový interpretovaný jazyk. Medzi jeho základné vlastnosti patrí jednoduchá syntax, ktorá zlepšuje čitateľnosť. Výhodou je veľké množstvo knižníc slúžiacich na prácu s webovými aplikáciami, s vývojom hier ale aj databázami a mnoho ďalšími. Taktiež je multiplatformový, takže aplikácia naprogramovaná v tomto jazyku môže byť spustená na zariadeniach s rôznymi operačnými systémami bez potreby upravovať kód. Python je na rozdiel od staticky typovaných jazykov, kde je potrebné vopred deklarovať typy všetkých dát, typovaný dynamicky [1].

NetworkX je open-source knižnica pre jazyk Python, používaná najmä na vytváranie, manipuláciu a študovanie štruktúry, dynamiky a funkcií grafových štruktúr. Poskytuje veľké množstvo algoritmov na analýzu, ako sú vzdialenosti medzi uzlami, hľadanie najkratšej cesty, hľadanie najmenšieho cyklu a mnoho ďalších. Zaujímavosťou je, že vrcholom grafu môže byť čokoľvek, od textového reťazca až po obrázky [2].

iGraph je open-source knižnica poskytuje širokú škálu pokročilých algoritmov pre analýzu vlastností sietí. Táto knižnica dokáže efektívne pracovať aj s veľkými dátovými súbormi, podporuje import a export dát z rôznych formátov čo zjednodušuje prácu s reálnymi dátami. Avšak na rozdiel od knižnice NetworkX ktorá podporuje iba jazyk Python, iGraph je možné využiť aj pri programovaní v jazykoch ako je Python, C/C++, R a Mathematica [3].

Tkinter je open-source knižnica pre jazyk Python, určená predovšetkým na tvorbu používateľského rozhrania pre desktopové aplikácie. Vývojárom poskytuje množstvo nástrojov na vytváranie, manipuláciu a správu

grafických komponentov, ako sú napríklad tlačidlá alebo polia na zadávanie textu. Tkinter je schopný práce s viacvláknovým prostredím, čo umožňuje efektívne riadenie viacerých úloh súčasne. Je obľúbený hlavne vďaka jednoduchšej syntaxi a intuitívnemu používaniu [4].

1.3.2. Java technológie

Java je vysokoúrovňový, staticky typovaný a objektovo-orientovaný programovací jazyk. Tento jazyk obľúbený kvôli tomu, že ponúka veľké množstvo nástrojov na podporu automatizovaného upravovania, debugovania, testovania a nasadzovania. Je známy aj vďaka svojej robustnosti a bezpečnosti [5].

JGraphT je open-source knižnica pre jazyk Java, zameraná na manipuláciu s grafovými štruktúrami. Podobne ako NetworkX v Pythone, JGraphT umožňuje vytvárať a analyzovať grafy a ich vlastnosti. Knižnica je navrhnutá tak aby efektívna a ľahko rozširiteľná [6].

2. ŠPECIFIKÁCIA

3. NÁVRH

digarmy

4. IMPLEMENTÁCIA

5. TESTOVANIE

6. POUŽITÁ LITERATÚRA

- [1] <https://www.python.org/doc/>
- [2] <https://networkx.org/documentation/stable/>
- [3] <https://python.igraph.org/en/stable/index.html>
- [4] <https://docs.python.org/3/library/tkinter.html>
- [5] <https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>
- [6] <https://jgrapht.org/>
- [7]
- [8]
- [9]
- [10]
- [11]
- [12]
- [13]
- [14]
- [15]
- [16]
- [17]
- [18]
- [19]
- [20]
- [21]
- [22]
- [23]