

Univerzita Karlova
Přírodovědecká fakulta
Úvod do programování



Úloha č. 127
Procházení grafu metodou BFS
Václav Thám 3. ročník B-SGG
Chlum 2022

Zadání:

Z textového souboru načtete reprezentaci grafu G a uložte ji. Každá hrana grafu má ohodnocení = 1. Pro implementaci metody BFS použijte frontu. Prohledávání grafu začne ve Vámi zvoleném uzlu, a do textového souboru se uloží výsledný BF strom.

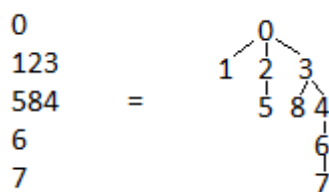
Rozbor problematiky

Formální definice

Prohledávání grafu metodou BFS neboli prohledávání do šířky je základní grafový algoritmus pro procházení grafu. Tento algoritmus systematicky prochází graf z libovolného uzlu a pro dočasné ukládání nenavštívených uzlů používá frontu. Díky ukládání do fronty dochází k procházení grafu po vlnách, kdy jsou uzly zpracovány v pořadí daném jejich vzdáleností od kořene.

Popis průběhu algoritmu

Na počátku algoritmu proběhne inicializace, poté se nastaví počáteční hodnoty pro jediný uzel, u kterého známe všechny informace. Rozběhne se cyklus, který běží, dokud není fronta prázdná (tzn. uzly se kterými pracujeme). Na počátku fronty vyjmeme uzel a pro všechny ostatní následníky budeme zjišťovat, zda byl bod již nalezen nebo prozkoumán. Když bude tato podmínka splněna, nastaví se informace pro tyto uzly (vzdálenost od počátku nastaví o jedna větší než uzel, kterého je tento uzel následníkem). Vzdálenost je větší právě o jedna, aby byla splněna podmínka, že je mezi uzly právě jedna hrana. Následně se pro tyto následníky nastaví předchůdce na hodnotu následníka uzlu. Po nastavení těchto vlastností je vložen do fronty. Když jsou prohledáni všichni následníci uzlu, uzel se zavře a pokračuje se na další.



Obrázek 1: testovací příklad a vizualizace stromu

Popis funkcí

class Vertex

Obsahuje statickou proměnou, která se o jeden zvyšuje s každým vrcholem, který přijme.

def __init__(self, label)

Konstruktor obsahující prázdný list sousedů, který se v průběhu zaplňuje přiřazováním.

def add_neighbour(self, v)

Zkontroluje, zda je vertex zapsaný v sousedech a když ne, tak mu dá append.

class Graph

def __init__(self, lines)

Funkce dostane v konstruktoru pole řádků (na každém indexu jeden řádek) a pro každý řádek rozdělí podle mezery. Třída bude mít v sobě uložené objekt s vrcholy. Pro každý řádek z řádků rozdělí mezerou, což vrátí pole o dvou prvcích. První index do Label1 a druhý do Label2. Pokud následně label1 nebude v uloženém seznamu, tak ho na pozici uloží pomocí třídy Vertex.

Podmínky pro label1 a label2 kontrolují, zda je již objekt v seznamu a když ne, tak ho vytvoří. Tyto podmínky se přeskočí, když se bude hrana vyskytovat již mezi uloženými labely. V tomto případě se hodí jako k label1 soused label2 a opačně též.

def read(input_name)

Čtecí funkce, která načte textový soubor zvolený uživatelem jako parametr. Nejprve fyzický soubor načte a poté rozdělí, aby na každém indexu byl jeden řádek. Funkce obsahuje 3 validační výjimky, kterými ověří, že soubor existuje, že k němu má přístup, a že je obecně validní pro použití programem

def breadth_first_search(graph)

Na začátku se inicializuje fronta, která bude prázdná, poté se inicializuje pole visited, které má zatím na všech indexech False (od indexu 0 po index velikosti grafu – počtu vrcholů). Následně se do fronty přidá první vertex a za index tohoto prvního vrcholu dáme True. Inicializuji výsledek, u kterého vím, že je samostatný. A vytvořím proměnou currentLevel, ta bude na začátku prázdná. Ve frontě se již nachází first vertex a tak mohu procházet, vyjmu vertex z fronty pomocí .pop, která bere první prvek, který jsem tam jako první vložil. Ověří, jestli je vertex v aktuální úrovni (currentLevel), což není, takže se podmínka přeskočí. For cyklus přidá souseda do fronty, pokud nebyl navštívený, jeho index změní na True (byl navštíven) a přidám ho do aktuálního levelu – sousedy prvního vrcholu. Pro sousedy se opakuje while dokud nejsou prozkoumáni všichni. Podmínka if přidává aktuální úroveň stromu

```
def save(file_name, tree)
```

Index úrovně v rozsahu délky souboru (stromu). Pro první vrchol je podmínka, aby nebyla mezera na začátku souboru, label se dá do stringu a uloží se, v případě více vrcholů skočí else, které mezi stringy přidá mezery. Pokud se index nerovná poslednímu indexu (je dosažena poslední úroveň), udělá nový řádek, odsadí a zapíše do souboru .txt.

Funkce ukládá vertexy po zjištěných levelech tak, že výsledný strom v textové souboru vypadá takto:

```
0
1 2 3
5 8 4
6
7
```

kdy jsou čísla rozdělena po úrovních.

Pseudokód:

```
import argparse
from cProfile import label
from graph import Graph

# Funkce pro čtení vstupního souboru
funkce čtení
    zkus otevřít vstupní soubor
        vrať vstupní soubor rozdělený na řádky
    výjimka soubor nebyl nalezen
        vytiskne chybu
        program skončí
    výjimka není přístup k souboru
        vytiskne chybu
        program skončí
    výjimka soubor není validní
        vytiskne chybu
        program skončí
# Hlavní funkce, která vytvoří seznam fronty a poté přidá vrcholy
funkce vyhledávání do šířky
    vytvoří frontu
    vytvoří navštívené a vloží do nich navštěvované vrcholy
    do fronty přidá první vrchol
    navštívené první vrchol označí true

    vytvoří výsledek a přiřadí do něj první vrchol
```

```

vytvoří proměnnou aktuální úroveň
# Procházení fronty a vyrovnávání
Dokud fronta
    vertex se rovná hodnota na indexu 0
    Když vertex je v aktuální úrovni
        výsledku přiřadí nakonec aktuální úroveň
    Pro souseda v vertex.sousedech
        Když není navštíven
            fronta přiřadí nakonec souseda
            souseda označí jako navštíveného
            aktuální úrovni přiřadí souseda

Vrátí výsledek
# uloží soubor za daných podmínek
funkce uložit
    Zkus otevřít výstupní soubor
        Pro index úrovně v rozsahu výsledného stromu
            vytvoří row_str a přiřadí ""

            Pro vertex ve výsledném stromu[index úrovně]
                Když je row_str rovno ""
                    row_str přiřadí string vertex.label
                Jinak
                    # Vloží mezeru mezi čísla v řádce
                    row_str += " " + string vertex.label

            Když index úrovně se nerovná nic z výsledného stromu -1
                row_str += '\n'

        do výstupního souboru zapiš řádku
# výjimky pro uložení
Výjimka nemá přístup
    vytiskne chybu
    ukončí program
Výjimka
    vytiskne chybu
    ukončí program

# Otevření vstupních a výstupních souborů jako argumentů
vytvoří argument i pro input
vytvoří argument o pro output
Když je dán argument pro input a zároveň argument pro output
    pomocí funkce čtení vytvoří proměnnou lines
    vytvoří proměnnou g z Graph(lines)
    vytvoří proměnnou počáteční vrchol, která je 0 index z listu vrcholů
    do proměnné výsledný strom vloží výsledky ze funkce vyhledávání do šířky
    funkce uložit pak uloží výsledky do souboru
Jinak
    vytiskne chybu, že jste nezadali požadované argumenty
    ukončí program

```

Vstupy a výstupy:

Vstup

Vstupem programu je textový soubor ve formátu číselných párů na jednotlivých řádcích.

Výstup

Výstupem programu je textový soubor, který má na řádcích reprezentované úrovně stromu.