

Univerzita Karlova
Přírodovědecká fakulta

Katedra aplikované geoinformatiky a kartografie



ÚVOD DO PROGRAMOVÁNÍ

Nalezení sjednocení dvou posloupností čísel
dokumentace

Lukáš Nekola
2. ročník, NKARTGD
Humpolec 2022

Obsah

1. Zadání.....	3
2. Rozbor problému.....	3
3. Existující algoritmy a možná provedení.....	3
4. Popis zvoleného algoritmu	3
5. Struktura programu (datové struktury, metody).....	4
6. Popis vstupních/výstupních dat.....	5
7. Problematická místa	5
8. Možná vylepšení	5
9. Zdroje	5

1. Zadání

Úkolem je vytvoření programu, který bude schopen ze dvou posloupností čísel zaznamenat jejich sjednocení. Tyto dvě posloupnosti čísel budou zadány uživatelem, který jednotlivá čísla oddělí mezerou. Na počátek si mohu definovat, že uživatel bude moci také zadávat čísla s desetinnými místy.

2. Rozbor problému

Nejprve je nezbytné rozebrat problematiku sjednocení, jakožto výrokového problému. Dle webového portálu Matematická logika (2022) se jedná o operaci pro spojení dvou množin. Výsledná množina tedy obsahuje takové prvky, které se vyskytují alespoň v jedné z počátečních množin. Další pravidlo říká, že výsledná množina prvků nesmí obsahovat duplicitní hodnoty. V rámci úkolu se bude jednat o sjednocení dvou posloupností čísel typu *float*. Nicméně abychom byly korektní, tento fakt by mohl být aplikovatelný na více způsobů sjednocení věnující se seznamům či jednorozměrným polím.

3. Existující algoritmy a možná provedení

Algoritmus pro výpočet průniku je velmi jednoduchý a můžeme konstatovat, že se zabývá spojením všech prvků dvou množin a následném odstranění redundantních hodnot. Z důvodu triviálního algoritmu bych se rád v této části věnoval způsobům v prostředí Python. Prvním způsobem je použití *operátoru* `|` pro sjednocení dvou množin. Druhý způsob se opírá o využití funkce *union* fungující na stejném principu jako zmíněný operátor (GeeksforGeeks, 2021). Avšak pro vytvoření programu v rámci úkolu byla vytvořena metoda fungující na naprogramovatelném sloučení dvou hodnot a odstranění duplicitních hodnot ve finální množině. Důvodem je snazší pochopení a popis funkčnosti.

4. Popis zvoleného algoritmu

Na počátku můžeme napsat, že algoritmus zjištění čísel můžeme dělit na dvě základní části. Před samotným zjištěním sjednocení dvou posloupností čísel je nezbytná úprava dat do zvoleného formátu. Jelikož uživatel smí zadat pouze posloupnost čísel, nikoliv jiných znaků textového řetězce, je nezbytné ošetření. Toto ošetření funguje na principu procházení jednotlivých znaků testového řetězce zadaného uživatelem. V případě

nepodporovaného znaku vypíše program upozornění, nicméně výpočet je dále prováděn, ačkoliv současný znak vynechá. Tento postup je proveden pro obě počáteční posloupnosti čísel.

Druhá část algoritmu se opírá o samotné sjednocení již ošetřených vstupů. Je zhotovena společná množina čísel, která je vytvořena spojením dvou seznamů. Po existenci konečné společné množiny je nezbytné odstranění redundantních hodnot. Jak si lze povšimnout, oproti ošetření vstupu se jedná o relativně snadný algoritmus. Jelikož výstup by měl být uživatelsky přívětivý, je nezbytné ošetření prázdné množiny a obeznámení uživatele s daným problémem.

5. Struktura programu (datové struktury, metody)

Samotný program je vyvinut o celkové délce 30 řádků včetně komentářů a odsazení pro snazší přehlednost. Součástí programu jsou dvě metody, kdy první z nich se stará o ošetření vstupu a druhá se stará o samotný výpočet sjednocení:

- a) První a zároveň nejsložitější část se zabývá již zmíněnou metodou pro ošetření vstupu. Tato metoda je volána pro oba vstupní textové řetězce. Před ošetřením je nezbytné uložení zadaného textového řetězce do seznamu na základě oddělovače, kterým je prázdný znak (mezera). Ošetření probíhá ve FOR cyklu, kdy jeho počet iterací závisí na počtu prvků již zmíněného pole. V samotném těle cyklu jsou postupně jednotlivé prvky seznamu ukládány do předem nadefinovaného nového prázdného seznamu, přičemž jejich ošetření spočívá v převodu na datový typ *float*. V případě nemožnosti převodu na daný datový typ upozorní program uživatele na chybný znak na konkrétní pozici.
- b) Druhá část se zabývá definicí metody pro samotné sjednocení, kdy vstupními parametry jsou ošetřené seznamy hodnot. V samotné metodě je nejdříve provedeno sjednocení prvků dvou seznamů a následně jsou jednotlivé prvky vzestupně uspořádány. Následuje odstranění redundantních hodnot. Tento proces je proveden funkcí *dict*, která funguje nad slovníky. Z tohoto důvodu byl seznam převeden na zmíněný slovník a následně zpětně uveden do původní podoby bez redundantních hodnot.

- c) V této části je uživatel dotázán na textové řetězce, které jsou následně ošetřeny první zmíněnou metodou.
- d) Poslední částí je výstup, který je podrobněji popsán ve druhé metodě. Výsledek je vytisknut do terminálu (příkazového řádku). V případě, že je výsledkem prázdná množina, uživatel je o tomto faktu informován za pomoci podmínky na nulovou délku výsledného seznamu.

6. Popis vstupních/výstupních dat

Za vstupní data považujeme 2 textové řetězce, které se zadávají do konzole, resp. příkazové řádky. Vstupní data by měla být v některém z číselných formátů. Výstup je opět vytisknut do konzole ve formátu *float*. Popisující program lze snadno upravit na možnost zadávání pouze celých čísel. Příklad vstupu a výstupu nalezneme v textovém dokumentu, který je přílohou této dokumentace.

7. Problematická místa

Za problematické místo můžeme považovat například výstup ve formátu *float*, který je méně přehledný než samotný formát pro definici celých čísel. Dočasné vytvoření slovníku pro odstranění redundantních hodnot je relativně náročnější částí a zcela jistě by šla tato část optimalizovat.

8. Možná vylepšení

Jak jsem uvedl v přechozí kapitole, byl by možný jiný způsob odstranění redundantních hodnot. Dle různorodých požadavků by bylo možné ukončení programu při špatně zadaných hodnotách. Jejich přeskočení a upozornění na chybu je součástí provedeného algoritmu. Za další vylepšení můžeme považovat volitelný počet vstupních množin.

9. Zdroje

GeeksforGeeks (2021): *Union() function in Python*

<https://www.geeksforgeeks.org/union-function-python/> (cit. 21.01.2022)

Matematická logika (2022): *Sjednocení množin*

<https://www2.karlin.mff.cuni.cz/~portal/logika/?page=sjed> (cit. 21.01.2022)