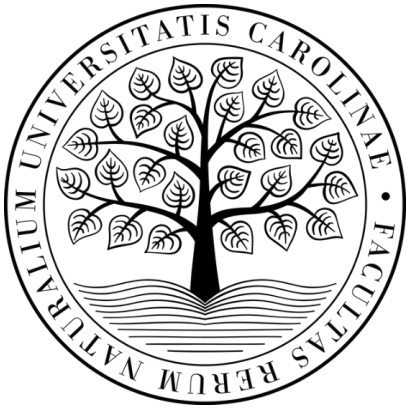
Univerzita Karlova

Přírodovědecká fakulta



ÚVOD DO PROGRAMOVÁNÍ

Odstranění duplicitních prvků z posloupnosti a sdělení jejich počtu

Petra Krsková

3. ročník Geografie a kartografie

Solnice 2022

Zadání

Na vstupu je nesetříděná posloupnost celých čísel. Nalezněte v posloupnosti duplicitní prvky, odstraňte je a sdělte jejich počet. Pokuste se tuto operaci učinit dostatečně rychle, aby byla metoda použitelná i pro dlouhé posloupnosti (v řádech stovek tisíc prvků). Vstupní data načtěte z textového souboru, výstup uložte také do textového souboru.

Součástí odevzdané úlohy bude zdrojový kód aplikace, vstupní/výstupní data a dokumentace se zadáním v rozsahu 4–5 stran ve formátu PDF obsahující následující:

- rozbor problému
- existující algoritmy
- popis zvoleného algoritmu
- struktura programu (datové struktury, metody,…)
- popis vstupních/výstupních dat
- problematická místa
- možná vylepšení

Aplikace bude považována za nefunkční, pokud:

- při zpracování dat dojde k pádu (runtime chyby, …)
- vrací špatné výsledky
- neřeší možné singulární případy

Rozbor problému

Posloupnost v matematice představuje sadu objektů, v níž se mohou jednotlivé objekty opakovat a záleží na jejich pořadí. Počet objektů, který může být konečný i nekonečný, je vyjádřen délkou posloupnosti. Posloupnost lze definovat jako funkční vztah mezi množinou přirozených čísel (pozice člena posloupnosti) a jejich obrazem (hodnota členu posloupnosti na dané pozici). Posloupnost bývá označována písmeny ve formě a_n, kde dolní index *n* označuje n-tý člen posloupnosti (Wikipedia 2022).

Existující algoritmy

Pro odstranění duplicitních prvků z posloupnosti existuje několik řešení. První možností je použití FOR cyklu, ve kterém jsou členy původního seznamu představující danou posloupnost čísel porovnávány s novým seznamem. Pokud se v něm nenachází, jsou do něj přiřazeny a na konci cyklu tak uživatel získá nový seznam, ve kterém se prvky z původního seznamu vyskytují právě jednou. Tento způsob je možné zkrátit i do podoby jednoho řádku za použití generátorové notace seznamu (list comprehension). Nejrozšířenější metodou pro odstranění duplicitních prvků je použití funkce *set().* Při jejím použití však dochází ke změně v pořadí členů posloupnosti, což je v rozporu s jedním ze základních pravidel posloupnosti zmíněným v předchozí kapitole. Nejrychlejší variantou je pak využití funkce *collections.OrderedDict.fromkeys()*, která na rozdíl od předchozí zachovává pořadí členů posloupnosti (GeeksforGeeks 2020; JournalDev 2022).

Popis zvoleného algoritmu

Algoritmy využívající vestavěné funkce popsané v předchozí kapitole jsou dobrým nástrojem pro rychlé odstranění duplicitních hodnot ze seznamu představující danou posloupnost čísel. Neumožňují však zaznamenávání smazaných duplicitních prvků a počtu jejich výskytu v posloupnosti. Z toho důvodu byl zvolen první zmíněný algoritmus založený na ukládání prvků do nového seznamu při jejich prvním výskytu v posloupnosti a zároveň zaznamenávání přeskočených duplicitních hodnot společně s jejich počtem do slovníku.

Pseudokód zvoleného algoritmu

sequence = vytvoření seznamu načtením hodnot ze vstupního souboru
final_sequence = vytvoření prázdného seznamu
duplicates = vytvoření prázdného slovníku
index = inicializace na hodnotu -1

cyklus *procházení prvků v seznamu sequence*
    zvýšení *hodnoty index o 1*
    odchytnutí chyby
        převést prvek na integer
    výjimka *v případě chybné hodnoty*
        vypsání chybové hlášky
        přeskočení na další prvek
    pokud *prvek je v seznamu final_sequence*
        vytvoření *klíče odpovídající prvku či zvýšení jeho hodnoty o 1*
    jinak
        přidání *prvku do seznamu final_sequence*
        zapsání *prvku do výstupního souboru*

Struktura programu

Program sestává ze 62 řádek včetně komentářů a odsazení a obsahuje dvě metody.

První metoda *SequenceFromFile* slouží pro otevření souboru a načtení vstupních dat. Pomocí *try* a *except* bloků je odchytáván špatně zadaný název souboru či cesta k jeho umístění, nedostatečné oprávnění pro přístup k souboru a další případné chyby spojené s otevíráním a čtením souboru. V případě chyby je do konzole vypsána chybová hláška definující problém a program skončí. Po otevření souboru je také zkontrolováno, zda obsahuje nějaká data a v případě prázdného souboru je opět vypsána chybová hláška a program skončí. Funkce vrací seznam hodnot typu *string* načtených ze souboru.

Druhá metoda *WriteToFile* slouží pro zápis výsledné posloupnosti čísel a duplicitních prvků do výstupního souboru. Podobně jako v předchozí metodě jsou zde pomocí *try* a *except* bloků ošetřeny možné chyby při otevírání souboru. Výstupní soubor je otevřen v módu *a*, který umožňuje připisování dat na aktuální konec souboru a nedochází tak k přepisu již existujících dat.

V další části programu je do proměnné *sequence* funkcí *SequenceFromFile* uložena posloupnost ze vstupního souboru ve formátu seznamu. Seznam *final_sequence* slouží pro ukládání jedinečných hodnot původní posloupnosti a slovník *duplicates* pak pro záznam odstraněných duplicitních prvků a jejich počtu. Proměnná *index* označující pořadí prvku v posloupnosti je zde inicializována na hodnotu -1.

Následuje hlavní část kódu ve formě *for* cyklu, ve kterém jsou postupně procházeny všechny prvky v seznamu *sequence.* Nejprve je hodnota proměnné *index* zvýšena o 1, aby odpovídala pořadí aktuálního prvku. Daný prvek je pak převeden na datový typ *integer* a uložen

do proměnné *element.* Pokud daný prvek není číslo a převod tak není možný, program vypíše chybovou hlášku a prvek přeskočí. Následně program pomocí podmínky *if* otestuje, zda se prvek nachází v seznamu *final_sequence.* Pokud ano, prvek je uložen jako klíč do slovníku *duplicates,* případně je zde jeho hodnota zvýšena o 1. Pokud ne, prvek je do daného seznamu přidán a zároveň i zapsán do výstupního souboru.

V poslední části programu je pak do výstupního souboru zapsán celkový počet odstraněných duplicitních prvků posloupnosti. Toho je docíleno součtem všech hodnot ve slovníku *duplicates.* Následně jsou pomocí *for* cyklu procházeny klíče a hodnoty v tomto slovníku, které odpovídají vymazaným duplicitním prvkům a jejich počtu, a funkcí *WriteToFile* jsou zapsány do výstupního souboru.

Vstupní a výstupní data

Program načítá vstupní data z textového souboru s názvem *input.txt*, který obsahuje na první řádce všechny celočíselné prvky posloupnosti oddělené mezerou. Při jiném názvu souboru je potřeba změnit kód na řádce 38, v závorce změnit název souboru či upravit cestu k jeho umístění. Pokud by byly od sebe prvky odděleny jiným znakem, je potřeba upravit kód programu na řádce 10 a to tak, že se daný znak zapíše do uvozovek do závorky funkce *split().*

Výslednou posloupnost a odstraněné duplicitní znaky včetně informace o tom, kolikrát byly vymazány, jsou zapisovány do textového souboru s názvem *output.txt*. Obdobně jako u vstupního souboru je v případě jiného názvu nutné upravit kód na řádkách 56, 59 a 61. Pro oddělení prvků výsledné posloupnosti jinak než mezerou je potřeba upravit kód na řádce 56 a to tak, že se v části *f"{element} "* mezera za složenou závorkou nahradí požadovaným znakem.

Problematická místa a možná vylepšení

Zdroje

GEEKSFORGEEKS (2020): Python – Ways to remove duplicates from list, https://www.geeksforgeeks.org/python-ways-to-remove-duplicates-from-list/ (5.2.2022).

JOURNALDEV (2022): Python Remove Duplicates from a List, https://www.journaldev.com/32742/python-remove-duplicates-from-list (5.2.2022).

WIKIPEDIA (2022): Sequence, https://en.wikipedia.org/wiki/Sequence (5.2.2022).