

Úvod do programování

Úlohy ke zkoušce

50: Odstranění duplicitních prvků z posloupnosti a sdělení jejich počtu

111: Dány dvě posloupnosti čísel, nalezení jejich sjednocení

Autor: Vít Šrámek

Obor a ročník: B-SGG, 3.

Semestr: ZS 2022/2023

1. Zadání

50: Odstranění duplicitních prvků z posloupnosti a sdělení jejich počtu:

Z textového souboru načtete vstupní data představující nesetříděné celočíselné hodnoty. Vámi vybranou třídící metodou data setřídíte a odstraníte opakující se prvky. Data zbavená duplicit uložíte do textového souboru a sdělíte počet duplicitních prvků.

111 Dány dvě posloupnosti čísel, nalezení jejich sjednocení:

Úloha je podobná té předcházející. Z textového souboru načtete vstupní data (setříděné celočíselné hodnoty) představující prvky obou množin. Vstupní množiny mohou být značně velké, použijte nějakou rychlou metodu, např. zatřídování (součást algoritmu merge sort). Realizujte sjednocení obou množin (žádný z prvků v ní nebude uložen duplicitně), výstupní data uložte do textového souboru.

Obě úlohy řešte s využitím OOP.

2. Popis a rozbor problému

V úloze 50 nalezneme nahraná nesetříděná čísla, která je potřeba seřadit, odebrat duplicitu a vypsát počet duplicit. Program data načte. Každému číslu přiřadí počet výskytů v souboru a na základě toho spočítá počet duplicit. Zároveň u toho odstraní duplicitní prvky – pokud již v souboru jsou, nevypíše je znovu.

V úkolu 111 řešíme obdobný problém, avšak mezi dvěma vstupními soubory. Program data načte, seřadí podle velikosti od nejmenšího po největší a vyřadí duplicitní prvky. Metoda řazení je inspirována metodou Merge Sort.

3. Popisy algoritmů formálním jazykem

50: Odstranění duplicitních prvků z posloupnosti a sdělení jejich počtu:

Pro výpis neduplicitních hodnot do výstupního souboru je použit tento kód, který číslo zapíše, nebo nezapíše na základě ověření jeho přítomnosti v seznamu. Nejprve je načten první prvek a dále je kontrolována přítomnost dalších prvků v seznamu. Proměnná counter slouží k nahrání první proměnné, pokud je 0, nahraje první hodnotu a dále probíhá kontrola vstupů. K té slouží proměnná in_list, pokud je 0, číslo se nahraje do seznamu, pokud je 1, číslo již v seznamu je a znovu se nenahraje.

```
#vypise neduplicitni cisla
def __init__(self, data):
    self.original_value = data
    counter = 0
    for i in data:
        val = Char(i)
        in_list = 0
        if counter == 0:
            self.data_without_duplicits.append(val)
        else:
            for j in self.data_without_duplicits:
                if i == j.value:
                    j.increment()
                    in_list = 1
                    self.duplicities += 1
            if in_list == 0:
                self.data_without_duplicits.append(val)
        counter += 1
```

Seřazení: vypsaná čísla jsou seřazena pomocí modulu sorted() od nejnižší numerické hodnoty.

```
#seradi cisla podle numericke hodnoty
def sort(self):
    sorted_list = []
    self.sorted_list = sorted(self.data_without_duplicits, key=lambda v: int(v.getValue()))
    print("Sorted data: " + str(self.sorted_list))
    return self.sorted_list
```

111: Dány dvě posloupnosti čísel, nalezení jejich sjednocení:

Seřazení čísel a odstranění duplicit: tato část kódu je inspirovaná metodou MergeSort. Čísla ze vstupních souboru jsou jednotlivě procházena (pomocí proměnné pointer1 a 2) a porovnávána. Zapsáno je vždy to menší, tím dojde k seřazení vstupních souborů. Zároveň je kontrolována a odstraněna duplicita.

```
#seradi cisla a odstrani duplicitu
for i in range(len_total):
    #vypise prebyvajici cisla z druhého souboru
    if (pointer1 == len1 - 1):
        for j in range(len2 - pointer2):
            if (value2[pointer2 + j]) not in self.sorted_number:
                if int(value2[pointer2 + j]) < int(value1[pointer1]):
                    self.sorted_number.append(value2[pointer2 + j])
                elif int(value2[pointer2 + j]) > int(value1[pointer1]):
                    self.sorted_number.append(value1[pointer1])
                for i in range(len2 - pointer2 - j):
                    if (value2[pointer2 + j]) not in self.sorted_number:
                        self.sorted_number.append(value2[pointer2 + j + i])
            print("End at value1")
            break
    #vypise prebyvajici cisla z prvního souboru
    elif (pointer2 == len2 - 1):
        for j in range(len1 - pointer1):
            if (value1[pointer1 + j]) not in self.sorted_number:
                if int(value1[pointer1 + j]) < int(value2[pointer2]):
                    self.sorted_number.append(value1[pointer1 + j])
                elif int(value1[pointer1 + j]) > int(value2[pointer2]):
                    self.sorted_number.append(value2[pointer2])
                for i in range(len1 - pointer1 - j):
                    if (value1[pointer1 + j]) not in self.sorted_number:
                        self.sorted_number.append(value1[pointer1 + j + i])
            print("End at value2")
            break
    #postupne seradi cisla podle velikosti a zapise je
    if (int(value1[pointer1]) > int(value2[pointer2])) and (int(value2[pointer2])) not in self.sorted_number:
        self.sorted_number.append(value2[pointer2])
        pointer2 = pointer2 + 1
    elif (int(value1[pointer1]) < int(value2[pointer2])) and (int(value1[pointer1])) not in self.sorted_number:
        self.sorted_number.append(value1[pointer1])
        pointer1 = pointer1 + 1
    elif (int(value1[pointer1]) == int(value2[pointer2])) and (int(value1[pointer1])) not in self.sorted_number:
        self.sorted_number.append(value1[pointer1])
        pointer1 = pointer1 + 1
        pointer2 = pointer2 + 1
    i = i + 1
```

4. Problematické situace a jejich rozbor

V úkolu 50 bylo třeba zajistit, abys se čísla nahraná jako objekty neřadila „abecedně“, tj. stylem 1,11,2,35,4, jak se dělo při jejich načtení z textového souboru. Opravy bylo dosaženo načtením „číselných“ verzí vstupu do řadičího programu za pomoci příkazu int().

V úkolu 111 byla prvotní verze řadičího kódu náchylná ke špatnému zařazení poslední hodnoty. Prvně byl po dosažení poslední hodnoty automaticky vypsán zbytek množiny druhé, nehledě na velikost čísel. To bylo ošetřeno přidáním další podmínky před zápis hodnoty.

5. Vstupní data, formát vstupních dat, popis

Vstupní data musí být nahrána v textovém souboru, který bude uložen ve stejné složce, jako samotný kód. Musí se jednat o celočíselné hodnoty. Zapsané musí být ve formátu jedno číslo jeden řádek. Pro úkol 111 musí být jednotlivé vstupní soubory již seřazené od nejmenšího po největší číslo. V neupravené verzi se vstupní soubor pro úkol 50 jmenuje vstup.txt a pro úkol 111 vstup2.txt a vstup3.txt.

6. Výstupní data, formát výstupních dat, popis

Výstupní data oba programy vypíší jako textový soubor. Čísla budou seřazena od nejmenšího k největšímu. Na každém řádku bude jedno číslo. V neupravené verzi programu se výstupy jmenují vystup1.txt pro úkol 50 a vystup2.txt pro úkol 111.

7. Dokumentace

50: Odstranění duplicitních prvků z posloupnosti a sdělení jejich počtu:
Základní informace:

Program nahraje zadaný vstupní soubor (v tuto chvíli vstup.txt). Spočítá počet duplicit a vypíše pouze neduplicitní, seřazená čísla do výstupního souboru (v tuto chvíli vystup1.txt) a zároveň do terminálu vypíše nahraná čísla a výsledky včetně počtu výskytů jednotlivých čísel a počtu duplicit. Program je neinteraktivní, stačí nahrát (případně zadat jména) vstupní a výstupní soubory a program spustit. Pokud program najde nějakou chybu (např. nenahraný vstupní soubor apod.) uživatele na to upozorní a skončí. Program si poradí s chybnými daty jako např. písmena, speciální znaky apod. a pracuje pouze s čísly

Uživatelská dokumentace:

Nahraj vstupní data ve správném formátu do stejné složky, ve které máš uložený kód. Potom stačí program spustit. Pokud program při načítání dat narazí na nějakou chybu, upozorní tě na to. Chybu oprav a program spustí znova. Pokud program chybu nenajde, nebo jsi ji opravil, stačí chvíli počkat a program ti do výstupního souboru vypíše výsledky. A to je všechno. Gratuluji, máš hotovo!

111: Dány dvě posloupnosti čísel, nalezení jejich průniku.

Základní informace

Program nahraje zadané vstupní textové soubory (v tuto chvíli vstup2.txt a vstup3.txt). Odstraní duplicitní hodnoty, čísla seřadí a vypíše je do výstupního textového souboru (v tuto chvíli vystup2.txt). Program je neinteraktivní, stačí nahrát (případně zadat jména) vstupní a výstupní soubory a program spustit. Program výsledky vypíše do výstupního souboru a zároveň vypisuje do terminálu nahraná čísla, který seznam skončil první a výsledky. Pokud program najde nějakou chybu (např. nenahraný vstupní soubor apod.) uživatele na to upozorní a skončí. Program si poradí s chybnými daty jako např. písmena, speciální znaky apod. a pracuje pouze s čísly.

Uživatelská dokumentace:

Nahraj vstupní data ve správném formátu do stejné složky, ve které máš uložený kód. Potom stačí program spustit. Pokud program při načítání dat narazí na nějakou chybu, upozorní tě na to. Chybu oprav a program spustí znova. Pokud program chybu nenajde, nebo jsi ji opravil, stačí chvíli počkat a program ti do výstupního souboru vypíše výsledky. A to je všechno. Gratuluji, máš hotovo!

8. Závěr, možné či neřešené problémy, náměty na vylepšení

Zadané úkoly se podařilo splnit. V úkolu 111 je dalším možným vylepšením kódu do budoucna zpracování dat ze souborů, které jsou neseřazené. V současnosti program pracuje s předpokladem, že jsou data seřazená, což může vést ke špatným výsledkům při použití vadných dat.

9. Seznam zdrojů

<https://www.scaler.com/topics/merge-sort-in-python/>

<https://stackoverflow.com/questions/46235507/sorting-instances-of-an-object-by-numeric-values-python>

https://cs.wikipedia.org/wiki/%C5%98azen%C3%AD_sl%C4%8Dov%C3%A1n%C3%ADm

<https://stackoverflow.com/questions/4617034/how-can-i-open-multiple-files-using-with-open-in-python>