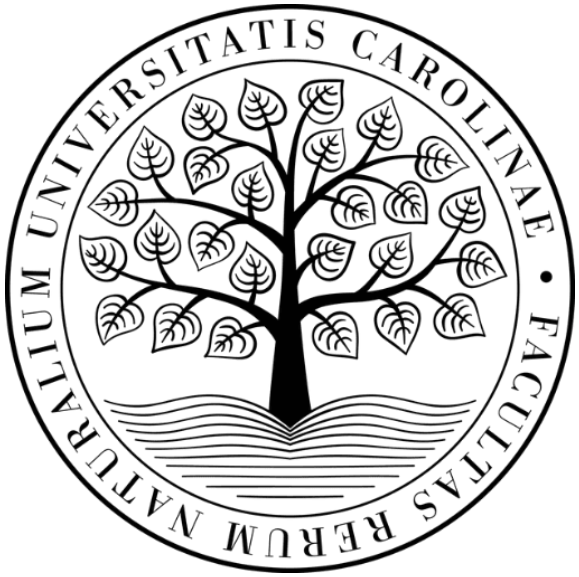
UNIVERZITA KARLOVA
PŘÍRODOVĚDECKÁ FAKULTA



ÚVOD DO PROGRAMOVÁNÍ

Úloha 92:
Výpočet modusu pro nesetříděnou posloupnost tvořenou *n* prvky

Anna Kožíšková

3. ročník, BGEKA

Mnichovice, 2022

Cílem úlohy je vypočítat modus pro nesetříděnou posloupnost o n prvcích. Posloupnost prvků bude uložená v seznamu (*list*) v proměnné, která bude součástí kódu. Uživatel, který bude chtít kód použít, tedy musí svá data sepsat do seznamu a uložit je přímo do kódu programu. Výsledný algoritmus bude umět procházet seznamy a nalézt jak číselné, tak textové prvky, které budou v souboru prvků nejčastěji zastoupeny.

Jelikož se úloha zabývá výpočtem modusu, je dobré říct si, co modus je. Modus je nejčastější hodnota ze souboru dat, nebo jiným opisem podle webové stránky Isibalo (2022): „… hodnota znaku s nejvyšší četností.“ Tedy pokud budu mít soubor dat o pěti prvcích x = [5,2,3,5,1], tak modus (M_x) pro tento soubor je 5, jelikož se v datech tato hodnota vyskytuje dvakrát (je nejčastější hodnotou v souboru). Pokud přidáme, do souboru další prvek o hodnotě 3, pak bude modus souboru 5 a 3. Je dobré poznamenat, že modus se dá také určit nejen pro číselné hodnoty, ale lze ho nalézt i pro slovní četnost. Pokud je dán soubor prvků y = [zelená, modrá, červená, modrá], pak M_y bude modrá.

Použitý algoritmus

Na stránkách https://github.com/koziskoa/Mode_calulation je k dispozici veřejný repozitář se souborem *code.py*, ve kterém se nachází výsledný kód programu.

V první části programu jsou definovány 2 proměnné typu list (*test*, *test2*). Proměnné jsou předvyplněné zejména proto, aby uživatel viděl příklad seznamu (jak s čísly, tak se slovy), který je schopen algoritmus zpracovat. Také je zde definován slovník (*freq_dict*), do kterého se budou během chodu programu ukládat znaky a jejich četnosti. Proměnná *mod_val* informuje o četnosti modusu v seznamu, tedy kolikrát se vyskytuje hodnota jednoho prvku (viz Obrázek 1).

Obrázek 1: Definování proměnných

```
4   # defining variables
5   test = [5,8,1,1,6,3,4,7,9,9,5,12,45,65,7,3,8,7,2,6,5,4,9,8,10] #=> 5,7,8,9
6   test2 = [1,3,9,45,1,2,7, "modrá", "modrá"] # => 1, modrá
7   freq_dict = {}
8   mod_val = 0 # frequency
```

Stěžejní část programu je založená na dvou cyklech. První cyklus (Obrázek 2) prochází seznam dat po jednotlivých prvcích. Každý nový prvek v seznamu se přidá do slovníku *freq_dict* a nastaví se mu hodnota 1, neboli, že se prvek vyskytuje v seznamu jednou. Pokud algoritmus narazí na prvek, který již byl do slovníku přidán, zvýší se četnost toho prvku o 1. Po dokončení iterace bude výsledkem slovník, jehož klíče jsou prvky ze seznamu a hodnoty jsou četnosti prvků.

Obrázek 2: Procházení seznamu a ukládání do slovníku

```
11  # iterrating trough list of unsorted data
12  for element in test:
13      if element not in freq_dict:
14          freq_dict[element] = 1
15      else:
16          freq_dict[element] += 1
```

Druhý cyklus (Obrázek 3) je tvořen dvěma vnořenými cykly. Celý cyklus proběhne 2x. Při prvním procházení se splní podmínka, že se *cycle* rovná nule. Vnořený cyklus v této podmínce bude procházet celý slovník *freq_dict* přes klíče a pokud je hodnota klíče (četnost) větší než *mod_val*, pak je aktuální procházená hodnota nejvyšší četností. V první fázi cyklu je tedy zjistějna informace o nejvyšší četnosti.

Rovnou by se dala zjistit i informace o modusu. Avšak pokud bude nejčastějších hodnot v seznamu více, pak si bude algoritmus pamatovat informaci jen o jednom prvku, a to o tom posledním nejčastějším, kdy se hodnota naposledy přepíše. Proto, je cyklus rozdělen na 2 vnořené cykly, kdy při druhém procházení již program ví informaci o nejvyšší četnosti. Při druhém procházení slovníku se hledají všechny klíče (prvky z původního seznamu), které dosáhly nevyšší hodnoty, která je uložená v proměnné *mod_val*. Při splnění této podmínky se vytiskne hodnota klíče, což je hledaný modus původního seznamu. Na úplný závěr se vytiskne informace o tom, kolikrát je prvek v seznamu zastoupen.

```
18  for cycle in range(2):
19      if cycle == 0: # interrating trough dictionary
20          for key in freq_dict:
21              if freq_dict[key] > mod_val:
22                  mod_val = freq_dict[key] #the highest frequency
23      else: # finds every modes in list
24          print(f"Mode in list:")
25          for key in freq_dict:
26              if freq_dict[key] == mod_val:
27                  print(f"    {key}")
```

Příklad výstupu:

```
Mode in list:
  1
frequency: 2
```

Alternativy programu

Jak již bylo naznačeno výše, druhý cyklus by mohl být jednodušší. V takovém případě však uživatel dostane správnou informaci jen v tehdy, když bude v seznamu právě jedena nejčastější hodnota.

Statistická knihovna Pythonu má k dispozici 2 funkce pro řešení nejčastější hodnoty v souboru prvků. První z nich *statistics.mode()* vrátí pouze jednu hodnotu ze souboru. Tato funkce ovšem neřeší výše popsaný problém. Pokud tedy uživatel potřebuje vědět všechny nejčastější hodnoty ze souboru, nabízí se použít funkci statistics.multimode(), která vrátí všechny nejčastější hodnoty.

Nedostatky programu

Program by mohl být vylepšen o metodu input, aby uživatel nemusel měnit v kódu obsah seznamu, nýbrž by se ho program dotázal na soubor prvků, které chce uživatel procházet. Uživatel by tedy nemusel měnit samotný kód programu

Zdroje

ISIBALO (2022): Modus a medián. https://isibalo.com/matematika/statistika/modus-a-median (cit. 10. 2. 2022)