Cvičenie na slovníky

Na zahriatie

- Vytvorte slovník, v ktorom bude heslá čísla od 0 do 10 namapované na ich druhé mocniny.
- Vytvorte slovník, v ktorom bude heslá čísla od 0 do 10 namapované na zoznam obsahujúci ich druhú, tretiu a štvrtú mocninu.

2 zoznamy <-> slovník

Naprogramujte funkciu, ktorá vám vytvorí z dvoch zoznamov slovník. Prvý zoznam sú heslá, druhý zoznam sú hodnoty. Napr:

```
keys = [1,2,3]
values = ['1', '2', '3']

dict_from_lists(keys, values) # -> {1: '1', 2: '2', 3: '3'}
```

Vytvorte funkciu opačnú, t.j., funkciu, ktorá vezme slovník a vráti dva zoznamy (heslá a hodnoty).

Otočenie slovníka

Naprogramujte funkciu, ktorá obráti slovník. Konkrétne, funkcia vytvorí nový slovník, kde heslá budú hodnoty s originálneho slovníka a hodnoty budú heslá. Napr:

```
orig = {1: 'a', 2: 'b', 3: 'c'}
invert_dict(orig) # -> {'a': 1, 'b': 2, 'c': 3}
```

Čo sa stane, pokiaľ sa niektoré hodnoty v originálnom slovníku opakujú?

Frekvenčná analýza

Vytvorte funkciu, ktorá vezme text a vráti slovník, ktorý mapuje slová v texte na to, koľkokrát sa slovo v texte nachádza, napr.:

```
# from "The Rime of the Ancient Mariner" by S. T. Coleridge
txt = "Alone, alone, all, all alone, alone on a wide wide sea!"

word_freq(txt)
# -> {'alone': 4, 'all': 2, 'on': 1, 'a': 1, 'wide': 2, 'sea': 1}
```

Zoskupovanie dát pomocou slovníka

Naprogramujte funkciu, ktorá vezme slovník slov a vytvorí slovník, kde každé heslo bude dĺžka slova a príslušná hodnota bude zoznam slov s tou dĺžkou. Napr:

```
words = ['jeden', 'dva', 'tri', 'štyri', 'pat', 'šesť']

len_dict(words)
# {
#   5: ['jeden', 'štyri'],
#   3: ['dva', 'tri', 'pat'],
#   4: ['šesť']
# }
```

Potom, funkciu upravte tak, aby brala dodatočný argument - funkciu. Táto funkcia bude určovať podľa čoho sa prvky zoznamu majú zoskupiť. Napr:

```
words = ['jeden', 'dva', 'tri', 'štyri', 'päť', 'šesť']

group_dict(words, len)   # -> to iste ako prva cast prikladu

def first_letter(x):
    return x[0]

group_dict(words, first_letter)
# ->
# {
#   'j': ['jeden'],
#   'd': ['dva'],
#   't': ['tri'],
#   'š': ['štyri', 'šesť'],
#   'p': ['päť']
# }
```

Množinové operácie

Dostanete zoznamy s menami študentov z troch predmetov. Vašou úlohou je nájsť, ktorí študenti mali:

- všetky 3 predmety
- aspoň 2 predmety
- presne 1 predmet

```
lst1 = ["Svätopluk", "Pribina", "Rastislav", "Otakar", "Gejza"]
lst2 = ["Gejza", "Svetlana", "Milena"]
lst3 = ["Pribina", "Gejza", "Rastislav"]
```

Deepcopy aj pre slovníky

Slovníky, rovnako ako zoznamy, môžu byť zanorené, napr:

```
d_outer = { 'a' : 1, 'b' : [1, 2] }
d_inner = { 'c' : 3.14 }
d_outer['inner'] = d_inner
```

Podobne ako v prípade zoznamov, keď chcete úplne nezávislú kópiu slovníku potrebujete implementovať *rekurzívnu* kópiu, ktorá vytvorí nové pod-slovníky (a pod-zoznamy) na každej úrovni zanorenia). V prípade normálnej (plytkej) kópie budú zanorené prvky na sebe závislé, napr (so slovníkmi nad):

```
d_outer['inner']['d'] = "Oh, no!"  # zmenim `d_outer`
print(d_inner)   # `d_inner` sa tiež zmení
```

Implementujte takúto kópiu. Uvažujte, že slovník môže obsahovať iné slovníky a zoznamy.

Poznámka 1: Vašu implementáciu môžete porovnať s funkciou `deepcopy` z balíčku `copy`. Vyskúšajte rôzne úrovne zanorenia (v príklade nad je len 1 úroveň, vaša funkcia by mala fungovať na najmenej 997 úrovní).

Poznámka 2: Snažte sa problému najprv porozumieť. Pre tento účel môžete použiť stránku pythontutor.com

Zoradený slovník

Od Pythonu 3.7 (ale prakticky 3.6) je slovník usporiadaný podľa poradia, v ktorom boli položky do slovníka vložené.

Napíšte funkciu, ktorá zoradí slovník (alebo vráti nový, zoradený) podľa hesiel - od najmenšej hodnoty k najväčšej. Napr:

```
d = { 'b' : -2, 'c' : 3, 'a' : 1 }
sort_dict_by_keys(d) # ->  {'a': 1, 'b': -2, 'c': 3}
```

Napíšte funkciu, ktorá zoradí slovník podľa hodnôt.

```
d = { 'b' : -2, 'c' : 3, 'a' : 1 }
sort_dict_by_vals(d) # ->  {'b': -2, 'a': 1, 'c': 3}
```

Slovník nôt

Hudobným tónom je možné priradiť frekvenciu. V klasickej hudobnej teórii hovorím napr. tónu "C0" zvuku s frekvenciou $\propto$ 16.35 Hz. Tón "o oktávu vyššie" je zase tón C ("C1") a má presne dvojnásobnú frekvenciu, t.j. $\propto$ 32.7 Hz.

Zostavte "kompletnú" (od "C0" do "C9") sadu párov *tón* - *frekvencia* z týchto dát:

```
some_tones = {
 'C5': 523.26,
 'C#4': 277.18,
 'D4': 293.66,
 'D#4': 311.13,
 'E4': 329.63,
 'F4': 349.23,
 'F#4': 369.99,
 'G6': 1568.0,
 'G#4': 415.3,
 'A4': 440.0,
 'B4': 466.16,
 'H5': 987.76,
}
```