FIKS 8. Ročník, 1. kolo, úloha 2

# Sponzoři – řešení

## Předpoklady

**Přiložené je praktické řešení v pythonu** *sponzori.py***. Ke správné funkci programu je potřeba do stejné složky vložit soubor se vstupem, jménem** *input.txt.* **Ve stejné složce program vytvoří/přepíše soubor** *output.txt* **s řešením. Řešení využívá dvě knihovny, tj.** *os* **a** *typing* **(obě jsou obsaženy v základní instalaci Pythonu). Kód úspěšně běžel na Windows 10 a 11, Python verze 3.9.4.**

## Teoretické řešení

1) Pomocí prvního řádku zadání si instrukce rozdělíme na id s jmény a sponzory se seznamy možných zvířat ke sponzorování. Páry id a jmen si necháme až na výstup, v algoritmech na přidělování budeme se zvířaty pracovat podle jejich id.
2) Nejprve si ale změníme přístup k datům, která máme a seznam, kde klíčem je jméno sponzora a hodnotou jsou id zvířat která je sponzor ochotný sponzorovat, přehodíme tak, aby klíčem bylo id zvířete a hodnotou seznam jmen potencionálních sponzorů. Tenhle seznam následně seřadíme vzestupně podle počtu možných sponzorů připadajícího k id, abychom při řešení preferovali zvířata s nejméně možnostmi, a tím našli sponzora co nejvíce zvířatům.
3) Při samotném přiřazovaní pak půjdeme podle indexu. Pro každé zvíře zvolíme první možnost v seznamu sponzorů, načež se tohoto jména, které jsme už přiřadili, zbavíme u všech dalších zvířat, abychom ho nepřiřadili dvakrát.
4) Pak už jen stačí vyměnit id za jména ve dvojicích id a sponzor a abecedně vypsat do souboru.

V tomto postupu dosahujeme časové náročnosti O(N^2) a paměťové náročnosti O(N)

## Popis kódu

*[V těchto závorkách se nachází řádky souboru sponzori.py relevantní k popisu]*

*Číslování částí koresponduje s teoretickým řešením*

1) [4-12] Nejprve otevřeme soubor a rozdělíme ho po řádcích do arraye *rawLines,* následně ještě první řádek (*rawLines* index 0) rozdělíme na dvě hodnoty, převedeme je na int a uložíme do arraye *instrukce* ***(-> instrukce[0] je počet zvířat v zadání, instrukce[1] je počet sponzorů v zadání)*** [14]. Pomocí čísel v něm pak můžeme z *rawLines* dostat pouze řádky jmen zvířat [16] a řádky sponzorů s jejich adepty [24]. Oba tyto seznamy řádků potom zpracujeme jako dictionaries. [18-22 & 26-31].
2) K přehození dat do vhodné formy si nejprve vytvoříme dictionary *zvireSponzori* kterému dáme *instrukce[0]* hodnot, čímž pokryjeme všechna id co se vyskytují v zadání, přiřadíme je jako klíče a jako hodnotu dáme prázdné arraye [33-36] abychom pak mohli použít append u připojování jmen [42]. Samotné přehazování pak mají na starost dva for loopy [38-42], první prochází seznam jmen a jejich adeptů, druhý v něm pak následně samotné adepty. Jméno z prvního loopu je pak v druhém přiřazováno ke klíči, kterým je id zvířete z druhého loopu. Tento dict je pak seřazen, nejprve pouze jako array klíčů [44] a poté jsou k těmto seřazeným klíčům zpět přiřazeny hodnoty z původního dictionary [46-49].
3) Přiřazování sponzora ke zvířeti probíhá jako for loop s počtem iterací *instrukce[0].* V loopu se nejprve vytvoří proměnné *id* a *sponzori* [56] které korespondují s prvním párem hodnot v seřazeném seznamu zvířat a potencionálních sponzorů. Do finálního dictionary jsou pak

vybírány páry id jako klíč s hodnotou *sponzori[0]* [58-59], pokud pro zvíře sponzor nezbyl, iterace vyústí v IndexError který je podchycen a pokračuje se dál bez zapsání do finálního seznamu. Na konci loopu je vždy zvíře se kterým jsme pracovali odstraněno ze seznamu [65 & 68], aby v další iteraci byl první pár ten další. Pokud se IndexError nestane, projde se celý seznam [61-63] a odstraní se sponzor který byl právě přiřazen ze seznamů potencionálních sponzorů ostatních zvířat.

4) Pak už máme dictionary *zvireSponzor,* který je v podstatě řešením úlohy. Ještě je pak jeho délka porovnána s počtem zvířat v zadání, abychom zjistili zda jsme našli sponzora pro každé z nich [73-76]. Ve zbytku kódu jsou pak id vyměněna za jména ze zadání [78-79], abecedně seřazena [81] a vypsána do stringu, který je zapsán do výstupního souboru [85-87]