

Lehký příklad — Šest stupňů odloučení

Šest stupňů odloučení je myšlenka, že všichni lidé jsou v průměru šest nebo méně sociálních kontaktů od sebe. Ve výsledku lze vytvořit řetězec „přítel přítele“, který spojí libovolné dva lidi v maximálně šesti krocích. Ověřte tuto myšlenku napsáním programu, který najde nejkratší cestu mezi dvěma lidmi v sociální síti.

Napište program **odlouceni.py**, který načte data sociální sítě ze souboru formátu JSON a nalezne nejkratší cestu mezi dvěma osobami. Podobnost jmen osob v síti a jejich přátel je čistě náhodná.

• Vstup:

- Program se spouští z příkazové řádky ve formátu:

```
python3 odlouceni.py <soubor.json> <start> <cil>
```

- **1. argument:** Název JSON souboru (např. `socialni_sit.json`). Soubor obsahuje slovník, kde klíčem je jméno osoby a hodnotou seznam přátel.
- **2. argument:** Jméno startovní osoby.
- **3. argument:** Jméno cílové osoby.
- Pokud jméno obsahuje mezery, musí být při spouštění v terminálu uzavřeno v uvozovkách (např. `"Jan Novák"`).

• Výstup:

- Pokud spojení existuje, vypište na standardní výstup:
 - * Jednu z možných cest ve formátu `Jméno -- Jméno --`
- Pokud spojení neexistuje (osoby jsou v oddělených komunitách), vypište `Cesta neexistuje`
- Pokud zadané jméno v síti vůbec není, vypište `Neexistující osoba`

• Poznámky k implementaci:

- Nezapomeňte soubor otevírat s kódováním UTF-8.
- **Načítání JSONu:** V Pythonu použijte vestavěnou knihovnu `json`. Data se načtou přímo do slovníku (dictionary).

```
import json
# ... uvnitř kódu ...
with open(nazev_souboru, 'r', encoding='utf-8') as f:
    data = json.load(f)
# Nyní je 'data' slovník: data["Jméno"] -> ["Přítel1", "Přítel2"]
```

Příklady

Příklad dat (obsah souboru `socialni_sit.json`)

```
{
  "Jan Novák": ["Petr Svoboda", "Eva Černá", "Martin Veselý"],
  "Petr Svoboda": ["Jan Novák", "Karel Pátek"],
  "Eva Černá": ["Jan Novák", "Karel Pátek"],
  "Karel Pátek": ["Petr Svoboda", "Eva Černá", "Boss"],
  "Martin Veselý": ["Jan Novák", "Lucie Modrá"],
  "Lucie Modrá": ["Martin Veselý", "Boss"],
  "Boss": ["Karel Pátek", "Lucie Modrá"],
  "Robinson": ["Pátek"],
  "Pátek": ["Robinson"]
}
```

Příklad 1 — Přímý soused (1 skok)

Spuštění programu:

```
python3 odlouceni.py socialni_sit.json "Jan Novák" "Petr Svoboda"
```

Výstup:

```
Jan Novák -- Petr Svoboda
```

Příklad 2 — Více možností (2 skoky)

Spuštění programu:

```
python3 odlouceni.py socialni_sit.json "Jan Novák" "Karel Pátek"
```

Výstup:

```
Jan Novák -- Petr Svoboda -- Karel Pátek
```

Komentář: Mezi Janem a Karlem existují dvě stejně dlouhé cesty (přes Petra nebo přes Evu). Program může vypsát kteroukoli z nich.

Příklad 3 — Vzdálenější spojení (3 skoky)

Spuštění programu:

```
python3 odlouceni.py socialni_sit.json "Jan Novák" "Boss"
```

Výstup:

```
Jan Novák -- Petr Svoboda -- Karel Pátek -- Boss
```

Komentář: Mezi Janem a Bossem existují dvě stejně dlouhé cesty (přes Petra nebo přes Evu a Karla). Program může vypsát kteroukoli z nich.

Příklad 4 — Cesta neexistuje (izolovaný ostrov)

Spuštění programu:

```
python3 odlouceni.py socialni_sit.json "Jan Novák" "Robinson"
```

Výstup:

```
Cesta neexistuje
```

Komentář: Robinson a Pátek tvoří vlastní skupinu, která není spojena s hlavní sítí.

Příklad 5 — Stejná osoba (Start = Cíl)

Spuštění programu:

```
python3 odlouceni.py socialni_sit.json "Jan Novák" "Jan Novák"
```

Výstup:

```
Jan Novák
```

Příklad 6 — Chyba ve jméně (neexistující osoba)

Spuštění programu:

```
python3 odlouceni.py socialni_sit.json "Jan Novák" "Gandalf"
```

Výstup:

Neexistující osoba