

# Řešení úlohy Sponzoři

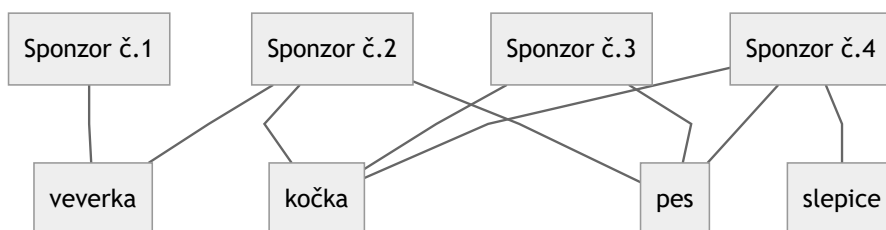
## 8. Ročník, 1. kolo

### Problém, který je nutno vyřešit

- Máme zadáno několik sponzorů, kteří sponzorují zvířata. Zadání nám říká přesně jaká zvířata může daný sponzor podporovat.
- Úkolem algoritmu je co nejrychleji najít ke každému zvířeti vlastního sponzora.

### Popis řešení

Tento problém si můžeme představit jako graf. To znamená, že si můžeme řešení graficky znázornit. Tady je jedna z nejjednodušších možností:



Graf (Příklad zadání)

### Zpracování dat

Při řešení budeme pracovat především s hranami grafu (Ty jsou v programu reprezentovány třídou Pair).

```
/** @var array<int $hash, Pair $pair> */  
public array $pairs = [];
```

Každý vrchol grafu (Sponzor / Zvíře) bude mít také svou třídu. Každému sponzorovi přidělíme unikátní id.

```
/** @var array<int $id, Sponsor $sponsor> */  
public array $sponsors = [];  
/** @var array<int $id, Animal $animals> */  
public array $animals = [];
```

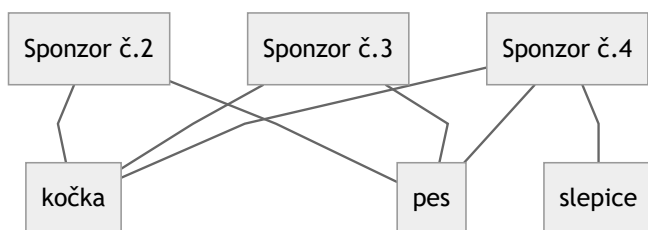
### Průběh řešení

Můj způsob řešení se skládá ze 3 metod:

#### 1. Metoda:

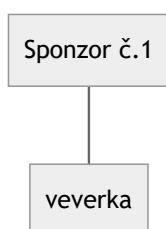
Ve většině případech máme možnost vyloučit **sponzora**, který má přiřazené jen **jedno zvíře** (např. v [grafu č.1](#) Sponzora č.1). Když tohoto sponzora společně s jeho zvířetem vyřadíme a předáme do finálního řešení, máme jistotu, že jsme spojili dvojici správně.

Po přesunutí první dvojice do pole konečného řešení musíme ze stávajících hran, se kterými pracujeme, odebrat všechny hrany které jakkoliv spojují zvíře nebo sponzora, kterého jsme právě přesunuli.



Graf (Krok 1)

Takto bude vypadat pole konečného řešení po prvním kroku:

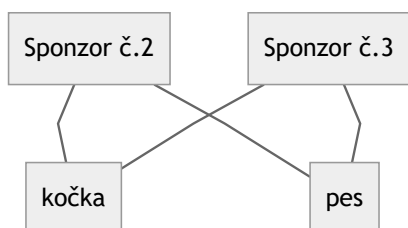


Řešení (Krok 1)

## 2. Metoda:

V případě, že pole neobsahuje sponzora, který má na sebe navázanou jen jednu hranu, můžeme (pokud to jde) aplikovat stejný postup na zvíře (v ukázkovém grafu na slepici)

Po této metodě bude graf vypadat takto:



Graf (Krok 2)

A pole konečného řešení takto:



Řešení (Krok 2)

### 3. Metoda:

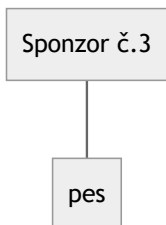
Když není možné aplikovat ani první, ani druhou metodu, musíme použít třetí

Vzhledem k tomu, že nelze aplikovat ani jedna z předchozích metod, každý sponzor má minimálně 2 zvířata a každé zvíře má minimálně 2 sponzory.

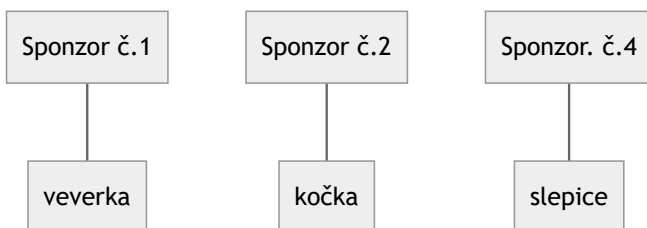
To znamená, že při spojení náhodné dvojice budeme mít nadále u každého vrcholu minimálně jednu hranu, kterou budeme moci využít k případnému připojení.

Ve třetí metodě tedy spojíme **jakéhokoliv sponzora** s **jakýmkoliv zvířetem**.

Po této metodě nám vznikne graf, který můžeme opět řešit první metodou.

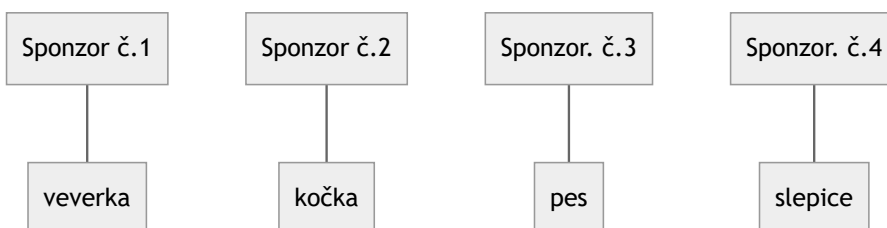


Graf (Krok 3)



Řešení (Krok 3)

Po opětovném použití 1. metody nám vznikne toto řešení.



Konečné řešení

## Spouštění programu

- Ke spuštění programu je potřeba min. verze [php 8.0](#)
- Program se spouští přes soubor `/Sponzoři/sponzori.php`
- Input je možné zadat buďto přes stdin (tuto metodu jsem nikdy netestoval), nebo přes textový soubor (toto je možné nastavit v souboru `/Sponzoři/sponzori.php`)