Řešení 2. úlohy 1. kola - Sponzoři

V této úloze musíme zvířatům přiřadit sponzory. Úkolem je přiřadit co nejvíce zvířatům sponzora, háček je ale v tom, že sponzoři jsou ochotní sponzorovat pouze některá zvířata. K řešní problému můžeme použít graf. V tomto grafu budou uzly sponzoři a žvířata a hrany budou představovat ochotu sponzora sponzorovat zvíře respektive možnost zvířete být sponzorováno sponzorem. My musíme zvířatům přiřadit sponzory, a to jsou dvě podmnožiny našeho grafu, takže se vlastně jedná o párování bipartitního grafu, a protože chceme přiřadit zvířat co nejvíce, tak se jedná mo maximální párování bipartitního grafu. Po rychlém googlování zjistíme, že se se problém dá převést na zjištění maximálního toku v síti. Proč to můžeme udělat? Do grafu si můžeme přidat dva další uzly, banku a zoo. Banku spojíme se všemi sponzory a zoo se všemi zvířaty, z banky tečou peníze a my je chceme dostat do zoo, co nejvíc samozřejmě. Z banky dostane každý sponzor jednu minci, a každé zvíře může předat zoo také jenom jednu minci. Proto když najdeme způsob největšího toku peněz budeme mít i maximální pokrytí zvířat sponzory. Algoritmus vymýšlet nemusíme, to už za nás udělali pánové Lester Randolph Ford ml. a Delbert Ray Fulkerson, použitím jejich Ford-Fulkersonova algoritmu. Naštěstí řešíme speciální případ, kde kapacita všech hran je 1, takže nemusíme řešit žádné zvyšování toku ani vracení zpět, protože nejkraší cesta, a také jediná možná bude vždy banka -> sponzor -> zvíře -> zoo. Algoritmus napíšu formou psudokódu, protože to je způsob, kterým se dá jakýkoli algoritmus pochopit nejjednoduššeji.

```
funkce muzePriradit(zvire){
    pro kazdeho sponzora{
        pokud sponzor je ochoten sponzorovat zvire a sponzor nebyl navstiven{
            nastav sponzora na navstiveneho
            pokud sponzor nebyl prirazen jinemu zvireti nebo muzePriradit(zvire
jiz prirazene sponzorovy){
                prirad sponzorovi zvire
                vrat Pravda
            }
        }
    }
    vrat Nepravda
}

funkce najdiParovani(){
    pro kazde zvire{
        nastav vsechny sponzory na nenavstivene
        muzePriradit(zvire)
    }
    vrat sponzory s prirazenym zviretem
}
```

Zbývá vyřešit časovou a paměťovou složitost. Funkce `muzePriradit` je sice rekurzivní, ale i v nejhorším případě bude zavolána `M`krát pak budou všichni sponzoři označeni za navštívené, takže má časovou složitost `O(M)`, funkce `najdiParovani` volá `muzePriradi` `N`krát, takže časová složitost té bude v nejhorším případě `O(M*N)`. Pro paměťovou složitost pak bude platit `O(M+N)`, protože máme uloženo `M` sponzorů a `N` zvířat.