Řešení 5. úlohy 2. kola - Prohlídkové okruhy

Úvod a porovnání algoritmů

V této úloze musíme najít, jestli existuje cesta mezi dvěmi křižovatkami. Na to můžeme použít několik algoritmů, většina z nich je založena na prohledávání do šířky (BFS), nebo prohledávání do hloubky (DFS). Vzhledem k tomu, že naším úkolem je jenom říct, jestli cesta existuje, tak nám stačí tyto základní algoritmy. Rozdíl mezi nimi je pouze v pořadí procházení uzlů. BFS si přidá všechny sousedy uzlu na konec fronty zpracovávaných uzlů. DFS přidá sousedy uzlu na začátek fronty. Kdyby byl náš graf strom, tak si to můžeme představit jako, že DFS z kmene dojde nejdřív uplně ne konec větví, až k listům a pak tohle opakuje na každém větvení, které po cestě potkalo. BFS, když přijde na větvení, tak prohlédne všechny sousední větvení, takže prochází graf ve vlnách a v každé vlně projde všechny větvení ve stejné vzdálenosti od kmene. Oba dva algoritmy mají stejnou časovou i paměťovou složitost v nejhorších případech, `O(M+N)`, algoritmus projde každý uzel právě jednou a každou hranu také právě jednou. Paměťová složitost, při ideálním naprogramování, závisí pouze na frontě, do které v nejhorším případě budeme muset nacpat všechny uzly kromě počátečního, proto je paměťová složitost obou `O(M-1)`.

Který algoritmus

Když jsou tedy oba algoritmy výpočetně stejně náročné, který vybrat? To záleží na vstupu. V tomto případě dostaneme graf, který obsahuje 3 cykly a z cyklů vedou slepé odbočky. Já předpokládám, že odbočky budou kratší než cykly, a proto bude rychlejší vybrat BFS.

Implementace

K jeho implementaci tedy: BFS začne v počátečním uzlu, přidá si všechny nenavštívené sousedy na konec fronty a nastaví je jako navštívené, a přesune se na další uzel ve frontě, zase přidá všechny nenavštívené sousedy uzlu na konec fronty a nastaví je jako navštívené, a přesune se na další, takhle pokračujeme, dokud buď nenajdem hledanou křižovatku, pak cesta existuje, nebo nedojdou uzly ve frontě, pak cesta neexistuje. Slovy pseudokódu

```
funkce najdiCestu(graf, start, cíl){
    fronta = prázdná fronta
    přidej start do fronty
    pro každý uzel ve frontě{
        pro každého souseda uzlu{
            pokud soused je cíl{
                vrať Cesta Existuje
            }
            pokud nebyl soused navštíven{
                přidej souseda na konec fronty
                nastav souseda jako navštíveného
            }
        }
    }
    vrať Cesta Neexistuje
}
```