

Teorie grafů – TGR

Úkol 2

Vypracoval: David Krénar

2a. Správa elektrické sítě

1 **Grafová struktura**

V daném úkolu byly transformátory zvoleny jako uzly a linky, které je propojují jako hrany grafu. Graf je reprezentován třídou s uzly a hranami. Hrany udržují informaci o vazbách mezi danými uzly. Použitou strukturou pro uchování těchto dat byly použity **seznamy** (list) pro uchovávání dat ze vstupního souboru.

2 **Použité algoritmy**

V rámci implementace byla použita datová struktura seznam. Seznamy s daty ze vstupního souboru tvoří seznamy počátečních a koncových uzlů (transformátorů) + seznam všech uzlů v grafu. Tyto seznamy jsou využity při testování grafu na smyčku a násobné hrany v grafu. Dále je zde třída „Graph“ – převzata z úkolu 2c, která testuje, zda se lze dostat z jednoho uzlu do dalších přes dané hrany (hledání mostů). Pokud najdeme nějakou hranu, která je mostem (odeberu ji a zvýší se počet komponent grafu), tak pak v síti nastane ERROR, jinak je síť OK.

3 **Zhodnocení**

Vzhledem k tomu, že to nebyl už první příklad, tak jsem měl již neměl problém se zpracováním vstupu. Problém však nastal, když jsem byl upozorněn na to, že nehledám úplný graf (hrany mezi každými uzly grafu), ale že se do uzlu mohu dostat i přes jiné uzly grafu. Tato skutečnost je samozřejmě uvedena v zadání v závorce „(i přes více transformátorů)“, ale při čtení zadání se nějak stalo, že tato informace byla přehlédnuta.

2b. Reset sítě

1 **Grafová struktura**

V daném úkolu byly transformátory zvoleny jako uzly a linky, které je propojují jako hrany grafu. Graf je reprezentován třídou s uzly a hranami. Hrany udržují informaci o vazbách mezi danými uzly. Použitou strukturou pro uchování těchto dat byly použity **seznamy** (list) pro uchovávání dat ze vstupního souboru.

2 **Použité algoritmy**

V rámci implementace byla použita datová struktura seznam. Seznamy s daty ze vstupního souboru tvoří seznamy počátečních a koncových uzlů (transformátorů) + seznam všech uzlů v grafu. Dále je zde třída „Graph“, která využívá implementaci „union by rank“, která je poté využita při implementaci **Kruskalova algoritmu** pro hledání minimální kostry grafu.

3 **Zhodnocení**

Problémem byla implementace Kruskalova algoritmu při hledání minimální kostry, ale jakmile se odstranili implementační errorry a podívání se na videa týkající se daného algoritmu, najít výsledek příkladu již problémem nebyl.

2c. Slabá místa sítě

1 **Grafová struktura**

V daném úkolu byly transformátory zvoleny jako uzly a linky, které je propojují jako hrany grafu. Graf je reprezentován třídou s uzly a hranami. Hrany udržují informaci o vazbách mezi danými uzly. Použitou strukturou pro uchování těchto dat byly použity **seznamy** (list) pro uchovávání dat ze vstupního souboru.

2 **Použité algoritmy**

V rámci implementace byla použita datová struktura seznam. Seznamy s daty ze vstupního souboru tvoří seznamy počátečních a koncových uzlů (transformátorů) + seznam všech uzlů v grafu. Dále je zde třída „Graph“, která testuje graf na hledání mostů a artikulací. Je využito přístupu prohledávání grafu do hloubky (DFS) a zjišťování, za je daná hrana mostem či uzlem artikulací.

3 **Zhodnocení**

Problémem byla implementace obou algoritmů (pro mosty i artikulace), ale jakmile se odstranili implementační chyby, tak zbytek příkladu již problémem nebyl.

2d. AVL Strom

1 **Grafová struktura**

V daném úkolu byla vytvořena třída „Node“ a „AVLTree“. Třída „Node“ uchovává informaci o uzlu ve stromu a třída „AVLTree“ reprezentuje AVL strom. Použitou strukturou pro uchování vstupních dat ze souboru byl použit **seznam** (list).

2 **Použité algoritmy**

V rámci implementace byly použity výše uvedené třídy, přičemž třída „Node“ uchovává informace o hodnotě uzlu a jeho možných potomcích. Třída „AVLTree“ uchovává informace o uzlech, o výšce stromu a zda je vyvážený.

3 **Zhodnocení**

Ze všech příkladů hodnotím tento jako nejtěžší (jak implementačně, tak rozumově). Problémem především byla implementace typů rotací v AVL stromu, aby zůstal strom vyvážený. Konečným problémem bylo vypsání seznamu čísel reprezentující strukturu AVL stromu pomocí metody procházení do šířky s přidávanými znaky „_“ pro chybějícího potomka, „|“ při rozlišování vrstev stromu a „“ při oddělování hodnot uzlů mezi levým a pravým potomkem. Tento výpis se mi nepodařilo implementovat, proto implementována metoda „display“, která ukazuje, jaká je podoba stromu (výpis se čte zleva po sloupcích (sloupec indikuje vrstvu stromu)).