

## 2a. Správa elektrické sítě

**2 body**

**Formulace problému:** Gratuluji, stali jste se správcem elektrické sítě v jedné nejmenované společnosti v jedné nejmenované zemi za devatero horami a sedmero ...

Vaším úkolem je kontrolovat stav sítě a zajistit, aby se elektřina dostala ke všem, co jsou připojení. Elektrická síť se skládá z transformátorů a linek, které je propojují. Mezi dvěma transformátory vede právě jedno spojení. Přípojky vedoucí od transformátorů ke koncovým uživatelům neuvažujeme.

Vámi vytvořený program dostane na vstupu aktuální stav sítě a musí kontrolovat, zdali mezi každými dvěma transformátory existuje právě jedno spojení (i přes více transformátorů).

**Název spustitelného souboru:** power

**Vstup:** Textový soubor obsahuje na každém řádku uvedenu jednu linku mezi dvěma transformátory, a její ztrátovost

```
T01 - T02: 15
T02 - T03: 10
T02 - T04: 2
T03 - T01: 18
```

**Výstup:** Pokud je síť v pořádku, vypište jednoduchou zprávu:

```
Stav site OK
```

Pokud ale zjistíte, že síť v pořádku není vypište:

```
Stav site ERROR
```

## 2b. Reset sítě

**2 body**

**Formulace problému:** Navazuje na předchozí úkol.

Během provozu se ukázalo, že občas je nutné síť restartovat, tedy přenastavit do optimálního stavu. Optimální stav znamená, že celková ztrátovost sítě je minimální. Síť se nejprve celá vypne a následně je zapojována jedna linka za druhou. Během tohoto procesu je nutné sledovat pořadí, ve kterém jsou linky zapojovány.

**Název spustitelného souboru:** reset

**Vstup:** Textový soubor, ve kterém jsou na každém řádku uvedena jedna linka mezi dvěma transformátory, její ztrátovost a stav. Z důvodu snazšího zpracování je formát vstupu stejný jako v předchozím úkolu.

```
T01 - T02: 15
T02 - T03: 10
T02 - T04: 2
T03 - T01: 18
```

**Výstup:** Výstupem algoritmu bude posloupnost linek tak, jak byly postupně zapojovány do sítě. Tato posloupnost závisí na algoritmu, který zvolíte. Výpis ukončete celkovou hodnotou ztrátovosti celé sítě:

```
T02 - T04: 2
T02 - T03: 10
T01 - T02: 15
Hodnoceni: 27
```

## 2c. Slabá místa sítě

**4 body**

**Formulace problému:** Navazuje na první úkol.

Během provozu se také ukázalo, že je potřeba monitorovat odolnost sítě vůči výpadkům. Z tohoto důvodu je dobré dopředu znát linky, za které není možné najít náhradu a transformátory, jejichž vyřazení by síť rozdělilo na dvě samostatné nespojitelné části.

**Název spustitelného souboru:** weakness

**Vstup:** Textový soubor obsahuje na každém řádku uvedenu jednu linku mezi dvěma transformátory, její ztrátovost a aktuální stav. Z důvodu snazšího zpracování je formát vstupu stejný jako v předchozím úkolu, ale ztrátovost.

```
T01 - T02: 15
T02 - T04: 2
T02 - T03: 10
T03 - T01: 18
T02 - T05: 11
```

**Výstup:** Výstupem algoritmu bude nejprve seznam linek, jejichž výpadek by vedl k rozpadu sítě a následně i seznam kritických transformátorů.

```
T02 - T04
T02 - T05
T02
```

## 2d. AVL Strom

**3 body**

**Formulace problému:** Na vstupu programu máte textový soubor s řadou číslí. Vaším úkolem je tento soubor postupně načítat a ukládat je do AVL stromu. V každém kroku vypište AVL strom.

**Název spustitelného souboru:** avltree

**Vstup:** Textový soubor, ve kterém je na každém řádku uvedeno jedno přirozené číslo. Může vypadat například takto:

```
5
15
9
5
155
```

**Výstup:** Výstupem bude seznam čísel reprezentující strukturu AVL stromu přepsanou pomocí metody procházení do šířky, kdy úrovně budu odděleny znakem | a prázdné místo (chybějící potomek) bude reprezentován znakem \_. Výpis bude končit poslední úrovní. Pro vstupní soubor z příkladu by pak výsledek programu mohl vypadat následovně:

```
5
5 | _, 15
9 | 5, 15
9 | 5, 15 | _, 5, _, _
9 | 5, 15 | _, 5, _, 155
```