

# Teorie grafů - TGR

## Úkol 1

Vypracoval: David Krénar

### 1a. Distribuční síť

#### 1 *Grafová struktura*

V daném úkolu byly sklady zvoleny jako uzly a dopravní prostředky jako hrany grafu. Graf je reprezentován seznamem uzlů a hran. Hrany udržují informaci o vazbách mezi danými uzly. Použitou strukturou pro uchování těchto dat byl **slovník** (dictionary).

#### 2 *Použité algoritmy*

Názvy skladů byly ukládány jako *key* a *value* tvořil počet výskytů daného skladu. Slovník byl zpracováván pomocí těchto parametrů, podmínek *if* a cyklů *for*. Hledanou vlastnost „nejvíce navštěvovaný“ jsem řešil pomocí hledání největšího stupně uzlů v grafu (maximální *value* ve slovníku u daného skladu). Hledanou vlastnost „existuje více spojení“ jsem řešil hledáním násobných hran v grafu. Hledanou vlastnost „zboží zpět do skladu“ jsem řešil hledáním smyček (cyklů) v grafu.

#### 3 *Zhodnocení*

Vzhledem k tomu, že to byl první úkol a první příklad, tak jsem měl problém se zpracováním vstupu. Zjistil jsem jaké funkce jazyk Python nabízí, které se dají k tomuto účely využít. Naučil jsem se uvažovat dat grafy z více pohledů a ne vždy první řešení vedlo k cíli.

### 1b. Rozdělení lidí

#### 1 *Grafová struktura*

V daném úkolu byly lidé zvoleni jako uzly a vlastnost „že se znají“ jako hrany grafu. Graf je reprezentován seznamem uzlů a hran. Použitá struktura pro uchování uzlů byl **seznam** (list). Dále byla použita struktura **množina** (set), pro redukci duplicitních osob v seznamech.

#### 2 *Použité algoritmy*

V rámci implementace byla použita datová struktura seznam, resp. dva seznamy. Jeden pro osoby, které někoho znají a druhý pro osoby, které jsou s někým kamarád. Práce se seznamy probíhala pomocí cyklů *for* a podmínek *if* pro testování, zda se znají a či lze osoby rozdělit do skupin.

#### 3 *Zhodnocení*

Problémem byl jak správně rozdělit osoby, tak aby se v dané skupině mezi sebou neznaly. Dané řešení není optimální vzhledem k počtu cyklů a podmínek a alokování místa v paměti.

## 1c. Silné týmy

### 1 Grafová struktura

V daném úkolu byly lidé zvoleni jako uzly a projekty, na kterých pracují jako hrany grafu. Graf je reprezentován seznamem uzlů a hran. Použitá struktura pro uchování uzlů byl **seznam** (list).

### 2 Použité algoritmy

V rámci implementace byla použita datová struktura seznam, resp. čtyři seznamy pro zpracovávání dat. Práce se seznamy probíhala pomocí cyklů *for* a podmínek *if* pro testování. Problém jsem identifikoval jako hledání kliky (ne všech klik, pouze největších, tak aby osoba byla vždy v jedné týmu). Při řešení byl využit **Bron-Kerboschův algoritmus** pro nalezení těchto klik.

### 3 Zhodnocení

Problémem bylo jak správně vytvořit matici ze vstupních dat, která poté byla zpracována daným algoritmem a jak vypsát pouze ty největší kliky, které jsou řešením úlohy. Tato úloha měla největší přínos. Dozvěděl jsem se o novém algoritmu v rámci NP-úplných grafových úloh. Pochopil jsem, co to vůbec je klika<sup>1</sup> v grafu, a jak se dají kliky najít.

---

<sup>1</sup>Klika grafu je každý maximální úplný podgraf grafu  $G$  (tj. každý úplný podgraf pro který platí, že žádný z jeho nadgrafů není úplný). Maximální číslo  $k$ , pro které existuje v  $G$  klika o  $k$  vrcholech nazýváme klikovostí grafu. Rozhodovací problém klikovosti grafu se ptá, zda-li v daném grafu existuje klika alespoň o  $k$  vrcholech. (zdroj: <https://www.algoritmy.net/article/7351/Kliky--%E2%86%92-Nezavisle-mnoziny>)