

Grafová štruktúra

Pre riešenie grafovej štruktúry som zvolil zviazaný zoznam susedov v kombinácii s objektovým prístupom (Python). Úroveň abstrakcie a dynamickosť štruktúry vyhovela požiadavkám jednotlivých podúloh a poskytla intuitívne rozhranie.

```
class Node:
```

```
    def __init__(self, name, value = None):  
        self.name = name  
        self.value = value  
        self.next = None
```

```
class Graph:
```

```
    def __init__(self):  
        self.graph = {}
```

Použité algoritmy

1. Sekvenčné prechádzanie jednotlivých vrcholov a inkrementovanie čítača.
2. Sekvenčné prechádzanie jednotlivých vrcholov a inkrementovanie čítača jednotlivých výskytov hrán, detekcia duplicít.
3. Spočítanie stupňov vrcholov algoritmom z podúlohy 1, detekcia stupňa rovná 0.
4. Sekvenčné prechádzanie hrán, detekcia zhody mena počiatočného uzlu a cieľového.
5. Kontrola hrany každý s každým okrem unárnych väzieb.
6. Kontrola hrany každý s každým okrem unárnych väzieb, pri existencii väzby jedným smerom, musí existovať aj druhým.
7. Počítanie výstupného a vstupného stupňa sekvenčným prechádzaním, zapojenie „Counter“ triedy pre najčastejšie výskyty.
8. Hĺbková kópia jedného grafu, následné prechádzanie druhého a pridávanie chýbajúcich spojení.

Zhodnocení

Riešenie úloh mi priblížilo pohľad na prácu s grafmi a ukázalo veľa možností na rôzne prístupy na dopracovanie sa k výsledku. Osviežilo algoritmické myslenie a tvorenie skriptov v Pythone za použitia objektového prístupu.