

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, dfnum = None, low = None):
```

```
        self.name = name
```

```
        self.value = value
```

```
        self.dfnum = dfnum
```

```
        self.low = low
```

```
        self.next = None
```

```
class Graph:
```

```
    def __init__(self):
```

```
        self.graph = {}
```

Použité algoritmy

1. Borůvkov algoritmus na nájdenie minimálnej kostry.
2. Algoritmus pre hľadanie artikulácií a mostov z prednášky.
3. Tarryho algoritmus na slepé prehľadávanie grafu.
4. Custom štruktúra pre vytvorenie AVL stromu, za použitia objektu `TreeNode` a `AvlTree`.

Zhodnocení

Riešenie úloh mi priblížilo pohľad na prácu s grafmi, konvertovanie z orientovaného na neorientovaný, prechody grafu do šírky a do hĺbky, hľadanie kostry grafu a pod. Osviežilo algoritmické myslenie a tvorenie skriptov v Pythone za použitia objektového prístupu.