

Úloha 2

1a - launch

- hľadanie minimálnej kostry grafu, použila som Kruskalov hladový algoritmus
- začala som zoradením hrán podľa ohodnotenia vzostupne, vytvorila som si zoznam označených uzlov, do ktorých som uzly postupne vkladala a zoznam dvojíc uzlov (čiže hrán), ktoré tvoria minimálnu kostru – tento zoznam bol aj výstupom metódy minSkelet()

1b – weakness

- úlohou bolo určiť artikulácie a mosty, na čo je potrebný algoritmus prehľadania grafu do hĺbky, ktorý som spracovala
- k riešeniu artikulácií a mostov som sa žiaľ nedopracovala

1c – escape

- v tejto úlohe šlo o neinformovaný prechod grafom
- v metóde som pomocou cyklu, ktorý skončil buď vtedy, keď som sa dostala do konečného uzlu unikovy_modul alebo keď sa counter dostal na číslo 50 (aby nedošlo k zacykleniu pri úlohách bez existujúcej unikovej cesty)
- prechádzala uzly a ich susedov, u ktorých som kontrolovala, či sú už v zozname navštívených uzlov a následne som si navštívené hrany ukladala do zoznamu wanderList, ktorý bol aj výstupom metódy wander()

1d – avltree

- úlohou bola implementácia AVL stromu
- vytvorila som si triedu TreeNode, ktorá má atribúty hodnota, ľavý potomok, pravý potomok a výška uzlu
- ďalej som si vytvorila triedu AVLTree, ktorá mala metódy insert na vkladanie hodnoty do stromu, ktorá volala metódy updateHeight na aktualizáciu výšky uzlu a rotate na prípadné vyvážovanie, ak to bolo potrebné (v tejto metóde sa kontroloval rozdiel hodnôt výšky ľavého a pravého podstromu a na základe toho bola prevedená buď rotácia vpravo alebo rotácia vľavo)
- nakoniec prevádzam pre-order výpis stromu, ktorý som ale nevedela sformátovať podľa vzoru v zadaní

Zhodnotenie

Zaujímavé úlohy, no trochu ma mätú tie chyby vo vzorových riešeniach. Ako najväčšie pozitívum beriem to, že som pochopila, ako funguje AVL strom (aj keď implementačne mi to žiaľ nefungovalo).