

Úloha 1

Využila som objektovú reprezentáciu grafu s jedinou triedou Graph. Dáta, ktoré sa majú do grafu ukladať sú vyberané pomocou regulárnych výrazov. Pri inicializácii sa dáta zo štandardného vstupu ukladajú do zoznamu (buď zoznam jednotlivých uzlov grafu, resp. zoznam skupín susediacich uzlov).

1a - akce

- neorientovaný graf, ktorý obsahuje 5 uzlov; zisťujem *stupeň uzlov*
- na tento príklad som zostavila slovník, kde kľúče sú jednotlivé uzly a hodnoty sú počet uzlov spojených s daným uzlom; pomocou tohto slovníka som zistila stupňu každého uzlu, zoradila som uzly od najväčšieho stupňa a nakoniec vypísala v požadovanom formáte

1b – sky_links

- orientovaný graf, ktorý obsahuje 5 uzlov; zisťujem, či je graf *multigraf* a hľadám *duplicitné hrany*
- na tento príklad som použila zoznam susedov (slovník, kde kľúče sú jednotlivé uzly a hodnoty sú zoznamy susedov kľúčového uzlu) a následne funkciu na nájdenie duplikátov v zoznamoch susedov; výstupom sú duplicitné uzly, vypísané v požadovanom formáte

1c – holidays

- orientovaný graf, ktorý obsahuje 7 uzlov; zisťujem, či má *každý uzol vstupnú/výstupnú hranu*
- načítala som si 2 zoznamy – 1 bol len zoznam uzlov (bez ohľadu na to, či má nejaké hrany) a druhý zoznam dvojíc uzlov, ktoré sú prepojené hranou (z druhého zoznamu som vytvorila zoznam susedov (slovník)); následne som si uložila do nového zoznamu len kľúče zo zoznamu susedov a tieto 2 zoznamy som porovnávala a vypísala som uzly, ktoré sa nachádzali v zozname uzlov a nenachádzali v zozname susedov

1d – wander

- orientovaný graf, ktorý obsahuje 6 uzlov; zisťujem, či graf obsahuje *slučku*
- v zozname susedov som zisťovala, či kľúč je zhodný z hodnotou a ak áno, uložila som si daný uzol do zoznamu, ktorý som následne vypísala

1e – company

- neorientovaný graf, ktorý obsahuje 5 uzlov; zisťujem, či je graf *úplný*
- v zozname susedov som zisťovala, či počet prvkov hodnôt slovníka (počet susedov) je rovný počtu uzlov mínus 1; výstupom je hodnota True alebo False

1f – communication

- orientovaný graf, ktorý obsahuje 5 uzlov; zisťujem, či existuje ku každej hrane *opačná hrana*
- pomocou zoznamu susediacich uzlov som zistila, či sa v danom zozname nachádza aj opačný uzol (node[::-1]) a ak nie, uložila som si daný uzol do nového zoznamu, ktorý som následne vypísala v opačnom poradí v požadovanom formáte

1g – distribution

- orientovaný graf, ktorý obsahuje 5 uzlov; zisťujem *počet vstupných a výstupných uzlov*
- postupovala som podobne ako pri úlohe 1a s rozdielom orientácie grafu, výstupom je uzol s najväčšou hodnotou vstupného (import) a výstupného (export) uzlu

1h – travel

- 2 orientované grafy, ktoré obsahujú 3 uzly; chcem zjednotiť 2 grafy (to sa mi ale nepodarilo, keďže mi robil problém ten upper/lower case)

Na začiatku som dlho tápala, ako vôbec načítať vstupné dáta do grafu. Následne som mala trochu problém s načítaním orientovaných hrán do grafu + graf, ktorý má viac ako 2 uzly na jednom riadku. Pri poslednej úlohe som nedokázala spojiť 2 grafy do jedného, aby som zachovala malé/veľké písmená.