

Teorie grafů – 1.úkol

1a) V tomto úkolu se jednalo o největší počet známých kamarádů. Úkol jsem řešila pomocí stupně uzlu. Uzly se načítají jako řetězce struktury. Každý uzel má určitý index pomocí, kterého lze graf prohlížet. Tudíž jsem procházela list, kde jsem si odstranila duplicitní hodnoty a následně spočítala stupeň daného uzlu. Po použití funkce „count“ a „sorted“ jsem si seřadila výchozí hodnoty a vypsala první 3, které měly nejvyšší hodnotu.

1b) Tento úkol pojednává o redundanci měst. Tedy o tom, zda nejsou dvě po sobě jdoucí místa na více pozicích. Uzly se načítají jako řetězce struktury. Každý uzel má index, který určuje pozici uzlu. Zde jsem musela použít prohledávání přes ítou pozici. Přes cyklus ověřuji, zda konkrétní název není na té samé pozici a pokud ano, přidám do jiného slovníku pro výstup. Nakonec si projdu prvky slovníku a vypíši vstupy.

1c) Úkol holidays je opakem úkolu 1b a kombinací s řešením 1a. Pojednává se o místa, která nebyla navštívena zaměstnanci. Použila jsem tedy stejnou funkci jako v 1a, kde počítám stupeň uzlu. Pokud je tento stupeň vyhodnocen jako „0“, poté ho vypíši přes („“.join()), který mi dané uzly seřadí do požadovaného formátu.

1d) Tento úkol se zabývá problematikou smyček, kde hledáme konkrétní uzel/uzly z kterých vedou cesty zpět do toho samého uzlu. Opět zde použiji prohledávání před ítou pozici, kde porovnávám, zda graf s daným uzlem se nerovná tomu samému grafu s daným uzlem +1, pokud je tomu tak, připsu uzel na výstup.

1e) Tento graf pojednává o úplnosti. Musela jsem tedy zjistit, zda konkrétní uzel má vazby ve všech ostatních. Tudíž jsem prošla všechny uzly grafu skrze pořadí na pozicích [i-1;i;i+1], jelikož jsou důležité vazby pro oba dva směry. Po zjištění polohy uzlu vedle něj, jsem uzel přidala do graph_listu. Odstranila jsem duplicitní hodnoty, které se ve výsledcích vyskytovali a vypsala jsem jejich hodnotu. Na konci jsem vypočítala hodnotu uzlu a pokud se rovnal max(počtu) v grafu, byl úplný, pokud chyběly hodnoty vypsala jsem „false“. Toto lze řešit také pro vzorec $(n*(n-1))/2$. Kde si spočítám všechny hrany a porovná se vzorcem.

1f) Tento úkol jsem se snažila řešit skrze procházení grafu, kde jsem si prošla první sloupec a spočítala jsem si jeho výskyt a totéž jsem udělala pro hodnoty druhého sloupce. Jelikož se ale v hodnotách vyskytují duplicity, snažila jsem se tyto hodnoty odstranit. Bohužel výpis těchto hodnot mi vyšel nesprávný

1g) Zde v tomto úkolu jsem si graf rozdělala na dva podgrafy. Tyto grafy jsem procházela podle indexů a počítala jsem počet výskytů uzlu. Po dokončení cyklu jsem si konkrétní uzly spočítala a vypsala jsem je pomocí funkce „sorted“ kde jsem je seřadila podle velikosti a následně vypsala s nejvyšší hodnotou.

1h) Poslední úkol byl zaměřený na sjednocení dvou grafů na základě spojení dvou firem. Tyto grafy byly v rozdílném formátu, proto bylo potřeba převést oba dva grafy do jednotného stylu lower(). Přechází vstupy, které budu porovnávat s hodnotami na výstupu. Pokud se dvojice ve výstupu nevyskytuje, přidám ji do seznamu, pokud ovšem ano, procházím cyklus dále.