

Globálne premenné **nie sú dovolené** .

Pri riešení úloh **môžete použiť** vlastné notebooky, STL.

Musíte použiť pripravený súbor ZadanieZP2.cpp, do ktorého doplníte kód.

1. Upravte funkciu loadFile (aj iné), ktorá načíta súbor data.dat, tak aby tento graf bol reprezentovaný ako matica susedov (adjacency matrix). (2 body)
2. Upravte funkcie getAverageInDegree a getAverageOutDegree (aj iné), ktoré vypočítajú priemerný stupeň uzlov tohoto grafu. (2 body)
3. Uprav funkciu getMaxInDegree a getMaxOutDegree (aj iné), ktoré nájdu uzol, ktorý má najväčší stupeň. (1 body)
4. Zvoľ najvhodnejší algoritmus na overenie dosiahnuteľnosti ľubovoľného uzla z každého iného uzla. Tento algoritmus použije maticu susedov ako reprezentáciu grafu. (5 body)

Štruktúra súboru data.dat : id trasy(1),odkiaľ(2),kam(3),ohodnotenie(4)

Bonus Naimplementuj metódu do triedy MHD, ktorá nájde časovo najkratšie (ohodnotenie) spojenie medzi dvoma zastávkami (uzlami) prestupy neuvažujeme. (5 bodov, avšak za semester je max. 20b)

Globálne premenné **nie sú dovolené** .

Pri riešení úloh **môžete použiť** vlastné notebooky, STL.

Musíte použiť pripravený súbor ZadanieZP2.cpp, do ktorého doplníte kód.

1. Upravte funkciu loadFile (aj iné), ktorá načíta súbor data.dat, tak aby tento graf bol reprezentovaný ako matica susedov (adjacency matrix). (2 body)
2. Upravte funkcie getAverageInDegree a getAverageOutDegree (aj iné), ktoré vypočítajú priemerný stupeň uzlov tohoto grafu. (2 body)
3. Uprav funkciu getMaxInDegree a getMaxOutDegree (aj iné), ktoré nájdu uzol, ktorý má najväčší stupeň. (1 body)
4. Zvoľ najvhodnejší algoritmus na overenie dosiahnuteľnosti ľubovoľného uzla z každého iného uzla. Tento algoritmus použije maticu susedov ako reprezentáciu grafu. (5 body)

Štruktúra súboru data.dat : id trasy(1),odkiaľ(2),kam(3),ohodnotenie(4)

Bonus Naimplementuj metódu do triedy MHD, ktorá nájde časovo najkratšie (ohodnotenie) spojenie medzi dvoma zastávkami (uzlami) prestupy neuvažujeme. (5 bodov, avšak za semester je max. 20b)