**ŽILINSKÁ UNIVERZITA V ŽILINE**

FAKULTA RIADENIA A INFORMATIKY

Optimalizácia sietí

MATEJ MAŽGÚT

**Semestrálna práca**

# Zadanie a popis metódy

Sieť: TN

Duálna heuristika: 2 – Algoritmus zväčšovania o najvýhodnejší uzol

Simulated Annealing: A – Výmena pod reťazcov dĺžky 4

Prvou časťou semestrálnej práce bolo naprogramovať riešenie úlohy obchodného cestujúceho s využitím heuristiky zväčšovania o najvýhodnejší uzol.

Ako prvé som si musel zostaviť základnú neprípustnú trasu. Prvý uzol trasy som vybral zo zoznamu všetkých nezaradených uzlov. Ďalej som vybral uzol, ktorý je od predchádzajúceho vybraného najviac vzdialený. A tretí som vybral rovnaký spôsobom ako druhý. Takto som vytvoril základnú neprípustnú trasu I1-> I2-> I3-> I1.

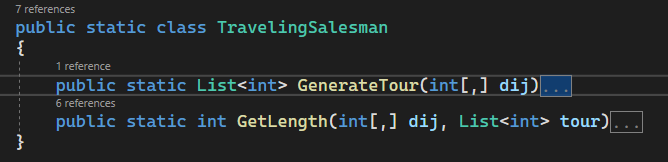
Ďalej bolo potrebne zaradiť všetky nezaradené uzly do trasy a to takým spôsobom, že vyberiem uzol, ktorý po dosadený do trasy zväčší dĺžku trasy najmenej. Takto treba dosadiť všetky nezaradený uzly a vrátiť sa späť na začiatok.

Druhou častou bolo výslednú prípustnú trasu zlepšiť pomocou metaheuristiky Simulated Annealing. Pričom okolie aktuálneho riešenia budem hľadať ako výmenu dvoch náhodných pod reťazcov dĺžky 4. Pre metaheuristiku bola zadaná počiatočná teplota t=10000, maximálny počet prechodov od prechodu k súčasnému riešenie u=40 a maximálny počet prechodov od zmeny teploty q=50.

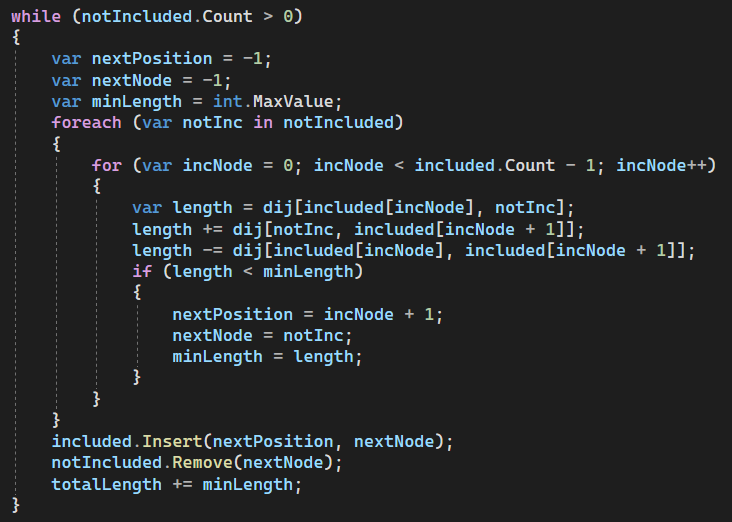
# Popis riešenia

Ako prvé bolo potrebné načítať maticu vzdialeností Dij. Tieto vzdialenosti som prečítal zo zadaného textového súboru TN a načítal do operačnej pamäte ako dvojrozmerné pole int-ov.

Po načítaní som si vytvoril triedu TravelingSalesman v ktorej sa nachádzajú dve metódy.

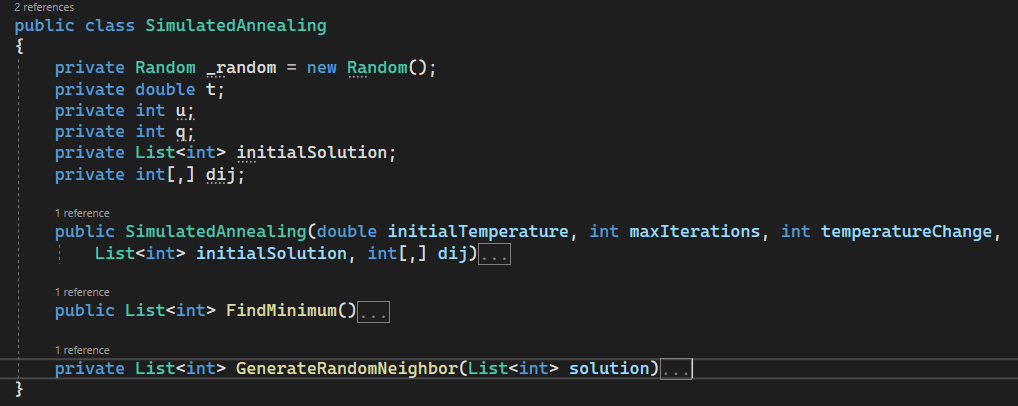


Prvá metóda slúži na vygenerovanie prípustnej cesty obchodného cestujúceho podľa zadania uvedeného vyššie. Jej návratovou hodnotou je list int-ov, ktorý predstavuje výslednú cestu. V danej metóde som ako prvé vygeneroval základnú neprípustnú trasu a ďalej som v cykle priraďoval do trasy nepriradené vrcholy na najvýhodnejšie miesto, až kým som ich nepriradil všetky. Cyklus môžete vidieť na obrázku.



Cyklus končí ak množina nezahrnutých uzlov je prázdna. Ak nie je, vyberie z tejto množiny vrchol, ktorý predĺži aktuálnu cestu o čo najmenej a dosadí ho na vhodné miesto.

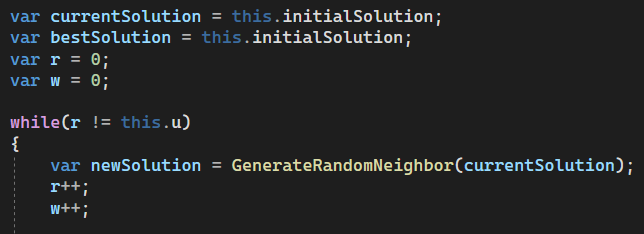
Druhá metóda slúži na vypočítanie dĺžky akejkoľvek trasy. Tú využívam pri výpise nájdenej trasy používateľovi a teda informovaní o jej dĺžke a tiež pri metóde Simulated Annealing.

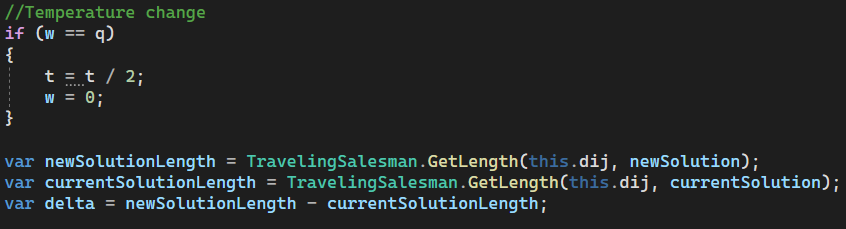
Pri zostavovaní metaheuristiky Simulated Annealing som si taktiež vytvoril samostatnú triedu, pre lepšiu prehľadnosť kódu. Jej štruktúra je zobrazená na nasledujúcom obrázku.

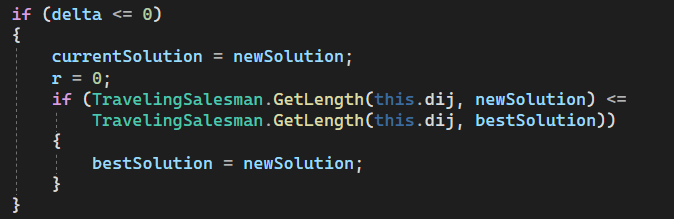
Táto trieda sa skladá už aj s konštruktora sa sady atribútov. Nachádzajú sa tu dve hlavné funkcie a to FindMinimum(), ktorá slúži na zlepšenie zadaného riešenia a tiež metóda GenerateRandomNeighbor(), ktorej návratovou hodnotou je cesta, kde boli náhodne vymenené dva pod reťazce dĺžky 4.

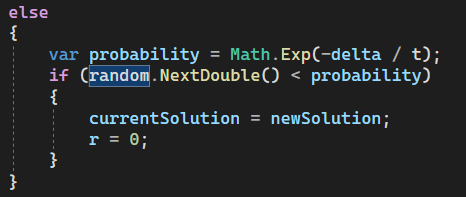
Jednotlivé názvy atribútov majú nasledovné významy:

* random slúži na vygenerovanie čísla od 0 po 1 pri rozhodovaní o prechode na nové riešenie a tiež na vygenerovanie náhodného čísla pri výmene pod reťazcov
* t je nutné uviesť pri vytváraní triedy a predstavuje počiatočnú teplotu
* u predstavuje maximálny počet prechodov od prechodu k aktuálnemu riešeniu
* q slúži pre maximálny počet prechodov od poslednej zmeny teploty
* initialSolution ako už z názvu napovedá slúži na uchovanie počiatočnej cesty
* dij predstavuje maticu vzdialenosti všetkých uzlov

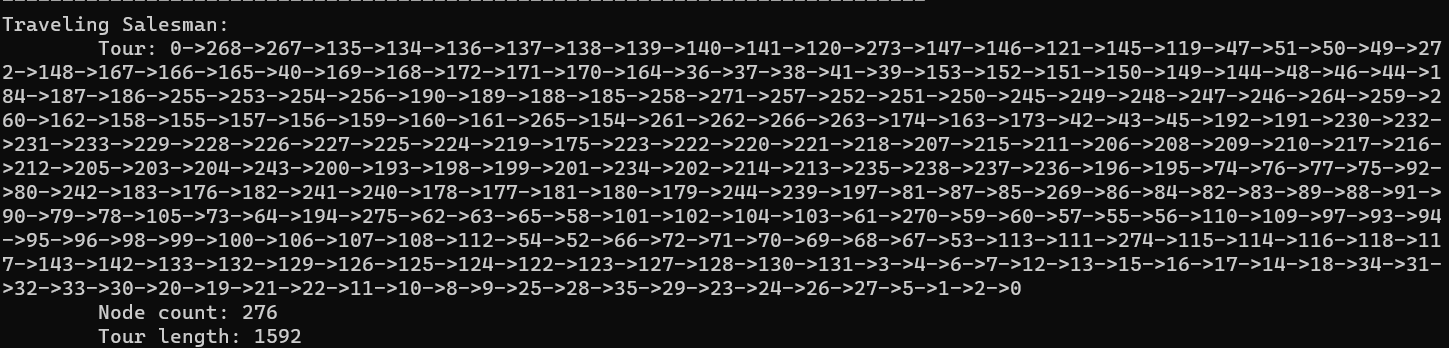
Samotný algoritmus na zlepšenie trasy sa skladá z niekoľkých častí. Ako prvé si definujem súčasne a najlepšie riešenie. A tiež premennú r, ktorá bude slúžiť pre zastavenie cyklu po určitom počte prechodov. Premennú w budem používať pre zmenu teploty po stanovenom počte prechodov. Obe premenne na začiatku cyklu navýšim a vygenerujem si nové riešenie, kde zamením dva reťazce dĺžky 4.

Ako ďalšie skontrolujem ci je potrebne znížiť teplou, ak áno znížim ju o polovicu a resetujem w. Ďalej zistím dĺžku nového riešenia, aktuálneho riešenia a do premennej delta si uložím ich rozdiel.

Ak tento rozdiel je menší ako nula a teda našlo sa lepšie riešenie ako je teraz, uložím si ho ako súčasne riešenie a resetujem premennú r pre skončenie cyklu. Ak toto riešenie je lepšie ako doteraz nájdene, tak si ho uložím aj do premennej s najlepším riešením.

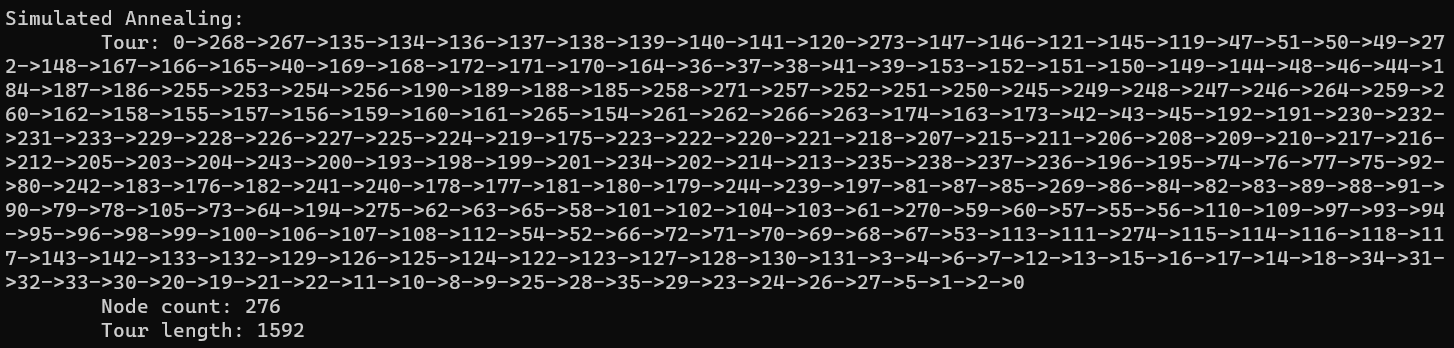
Ak je riešenie horšie ako súčasné, vygenerujem si pravdepodobnosť prechodu a ak táto pravdepodobnosť je väčšia ako náhodne vygenerované číslo, prechod k horšiemu riešeniu akceptujem a ponechám si ho ako súčasné riešenie.

# Výsledne riešenie

Výsledne riešenie pre súbor TN a úlohu obchodného cestujúceho vypíšem na konzolu a vypadá nasledovne.

Ako prvé je vypísaná výsledná cesta. Potom sa tu nachádza počet uzlov z ktorých cesta pozostáva a tiež jej výsledná dĺžka.

Pre úlohu Simulated Annealing je výpis podobný a je zobrazený nižšie.



# Záver

Úloha obchodné cestujúceho nám dala riešenie, kde dĺžka cesty bola rovná 1592. Pri metaheuristike Simulated Annealing sa cesta žiadnym spôsobom nezmenila a ani sa nezlepšila dĺžka cesty. To môže byť spôsobené tým, že metaheuristika nenašla lepšie riešenie a už prvotné riešenie nájdené duálnou heuristikou sa nachádza blízko optimálneho riešenia.