

Задача за търговския пътник

-1- Реализирани класове:

City - string name, int index - Изгражда град и му задава индекс, който по-късно ще използваме.

Relation - string city1, string city2, int costValue - Изгражда връзката между два града и цената на пътя между тях.

PathsMatrix - double** matrix, City* cities, Relation* relations, int size, string fileContent - Изгражда матрицата, която представя всички връзки между градовете, като на позиция index1(city1) и index2(city2) поставя цената на пътя между двата града.

HamiltonianCycle - Използване на Backtracking Algorithm, за да определим дали съществува път през всички градове, без да се повтарят, връщайки се в началният град.

GreedyApproachAlgorithm - Избраният за реализация Алгоритъм, представен в следващата точка.

-2- Реализиран Алгоритъм

Принцип на работа - **На всяка стъпка от пътуването посетете най-близкия непосетен град.**

Подобен на Алгоритъма на Дейкстра.

-3- Принцип на работа

Извличане на информацията от текстов файл, написана по начин, подходящ за запазване в .dot файл и визуализиране на връзките като граф.

Разпределяне на извлечената информация по градове и връзките им.

Създаване на таблица, представяща връзките между градовете, на база извлечената и разпределена информация.

Проверка е възможно да има път, минаващ през всички градове точно по веднъж и връщайки се обратно в началният град.

Ако е възможно, използвам избраният имплементиран алгоритъм, за да открия дължината на този път.