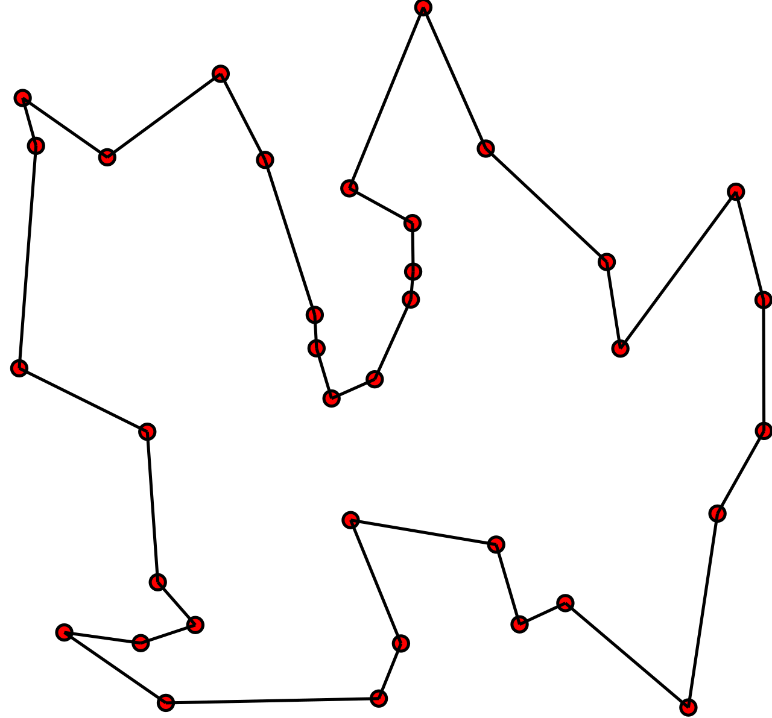
*УНИВЕРСИТЕТ ПО БИБЛИОТЕКОЗНАНИЕ И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ*

*ФАКУЛТЕТ „ИНФОРМАЦИОННИ НАУКИ“*

*Курс „Основи на програмирането“*

Анализ, решение и имплементация на задача „Бензиностанции“

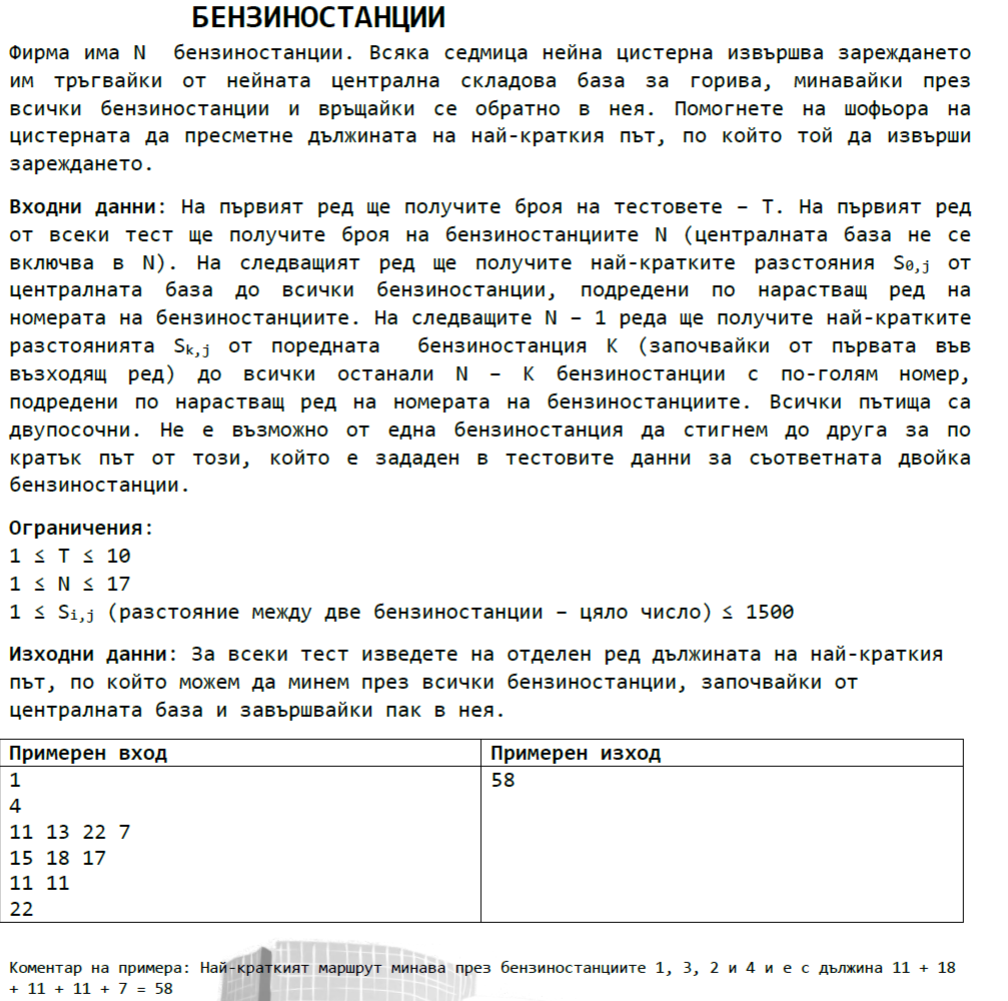
*Автор: Костадин Динков, 1-ви курс, направление ИКН, фак. № 46355з*



# Абстракт

Целта на този документ е да представи практическа имплементация на решение на задачата с „Бензиностанциите“, зададена като освобождаваща от изпит в курса „Основи на Програмирането“ ( УНИБИТ, ИКН, първи семестър). Проблема се анализира и се разглеждат няколко възможни подхода за решаването му, като метод с изчерпателно търсене и приблизителни решения. За самата имплементация на решението се използва алгоритъма на **Белман-Хелд-Карп** за класическия проблем с пътуващият търговец (**Traveling Salesman Problem**). Обясненията са насочени към хора с начален опит в програмирането и алгоритмите, но ще бъдат най-подходящи за тези, които сами са се опитали да намерят решение на задачата. Решението е имплементирано на C#, използвайки .Net Core технологията.

# Условие На Задачата

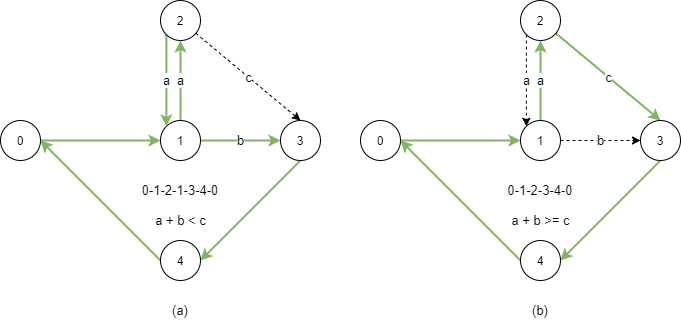


# Анализ

От условието на задачата е видимо, че тя е много прилича на класическата задача за пътуващия търговец (Traveling Salesman Problem – TSP). Като входен параметър имаме пълен, неориентиран граф. Пълен, защото имаме пътища (ребра) от една бензиностанция (връх) до всяка друга бензиностанция. Трябва да намерим най-краткия път от даден начален връх , така че този път да премине през всички върхове в графа и да се върне обратно в първия. Такъв път ще наричаме обиколка.

## С овторение или без повторение

За разлика от класическия TSP, в който имаме ограничението, че трябва да посетим всеки връх веднъж и само веднъж, нашата задача няма такова условие. Т.е. е възможно да имаме оптимален път, който посещава една бензиностанция повече от веднъж, като например 0-1-2-1-3-4. Това означава, че е пътят от 2-3 е по-дълъг отколкото сумата на пътищата от 2-1 и 1-3, така че за цистерната е по-добре да се върне от 2 до 1 и след това да стигне от 1 до 3, отколкото да стигне от 2 директно до 3. Това е демонстрирано на Фигура 1 (а). Но в нашият вариант на задачата имаме и допълнителното ограничение, че „не е възможно от една бензиностанция да стигнем до друга за по-кратък път от този, който е зададен в тестовите данни за дадената двойка“. Това означава, че примера по-горе не отговаря на условията, т.е. директният път от 2 до 3 винаги ще е по-кратък или равен на сумата на пътищата от 2-1 и 1-3 (Фигура 1(b)).



Фигура 1   
(а) Възможна обиколка, когато можем да повтаряме посещения на върхове.  
(b) Възможна обиколка, когато можем да повтаряме върхове и не съществува по-кратък път от директния между два върха.

От тук следва и е лесно доказуемо, че ако в нашата задача имаме оптимална обиколка с повторения, като например  
където са върхове от графа, а е разстоянието между два върха, то ако премахнем повторенията на , ще получим обиколка  
който е по-кратък или равен на първия, защото  
 Всъщност ще можем да премахнем повторенията на всеки връх и да получим обиколка без повторения, която е с по-малка или равна дължина на тази с повторенията. Единственото условие е върхът, който сме избрали за начало да се повтаря в края на обиколката, но можем да премахнем други негови повторения в средата. Като обобщение можем да кажем, че ако е дължината на оптимална обиколка, в която има повторения на върхове, а е дължината на такава, в която няма повторения то:

Какво означава този извод за нашия проблем? Това, че в условието не е дадено ограничение за броя на посещенията на един връх няма значение за крайният ни резултат. Обиколката без повторения винаги ще е по-кратка или равна на такава с повторения.

## Извод

От написаното до тук следва, че можем да приравним нашата задача на класически частен случай на TSP, а именно MTSP (metric TSP) където е валидно правилото за неравност в триъгълника

където e функция за разстояние между два върха.

Вече имаме насока, в която можем да направим проучванията си за намиране на решение на проблема. От една страна, TSP e един от най-изследваните и популярни проблеми в компютърните науки и за него съществуват различни решения и алгоритми. От друга страна, към датата на писането на този документ, не е известно бързо решение което да дава точен резултат. Под бързо се има предвид решение, което да дава резултат в полиномно време, т.е. времето за изпълнение е полиномна функция на размера на входните данни. Оказва се, че времето за намиране на точен резултат на нашият проблем не е полиномна, а експоненциална функция на броя на бензиностанциите. За да припомним, ако n е броя на бензиностанциите, то полиномна функция е например , или. Експоненциалната функция може да изглежда така .

Това което ни остава е да намерим най-подходящ алгоритъм, който да ни дава точен резултат при максимум 17 бензиностанции.

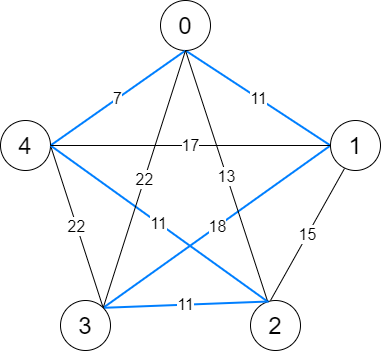
# Алгоритъм с Изчерпателно Търсене

Най-интуитивният начин за решение е да атакуваме проблема директно, чрез използването на brute-force (груба сила?) алгоритъм, т.е. като изброим (енумерираме) всички възможни обиколки, изчислим техните дължини и накрая изберем като резултат най-малката от тях. Ако приемем, че броят на градовете е N, то всички възможни обиколки от даден връх в графа са (факториел), a **различимите** са . Това е така защото една обиколка може да бъде направаена и в обратна посока.

|  |  |  |
| --- | --- | --- |
| Обиколка | Обиколка в обратна посока | Дължина |
| 0-1-2-3-4-0 | 0-4-3-2-1-0 | 11+15+11+22+7=66 |
| 0-1-3-2-4-0 | 0-4-2-3-1-0 | 11+18+11+11+7=58 |
| 0-1-3-4-2-0 | 0-2-4-3-1-0 | 11+18+22+11+13=75 |
| 0-1-2-4-3-0 | 0-3-4-2-1-0 | 11+15+11+22+13=72 |
| 0-1-4-3-2-0 | 0-2-3-4-1-0 | 11+17+22+11+13=74 |
| 0-1-4-2-3-0 | 0-3-2-4-1-0 | 11+17+11+11+22=72 |
| 0-2-1-3-4-0 | 0-4-3-1-2-0 | 13+15+18+22+7=75 |
| 0-2-1-4-3-0 | 0-3-4-1-2-0 | 13+15+17+22+22=89 |
| 0-2-3-1-4-0 | 0-4-1-3-2-0 | 13+11+18+17+7=66 |
| 0-2-4-1-3-0 | 0-3-1-4-2-0 | 13+11+17+18+22=81 |
| 0-3-1-2-4-0 | 0-4-2-1-3-0 | 22+18+15+11+7=73 |
| 0-3-2-1-4-0 | 0-4-1-2-3-0 | 22+11+15+17+7=72 |

Фигура 2.

Диаграма на граф с 5 върха и оптимална обиколка 0-1-3-2-4-0 с дължина 58. В таблицата са дадени всички възможни обиколки с начален връх 0, както и техните дължини.



На Фигура 2 е дадена диаграмата на примерния входен тест от заданието на проблема. В таблицата на Фигура 2 са изброени всички възможни пътища с начален връх 0, преминаващи през всички останали върхове и връщащи се обратно до началния връх. Виждаме, че броят на възможните обиколки е 24, което е точно :

Виждаме също, че една обиколка може да бъде направена и в обратна посока и това няма да промени нейната дължина. Например обиколката 0-1-2-3-4-0 с размер 66 е с еднаква дължина и в обратна посока 0-4-3-2-1-0. Това намалява размера на възможните решения на 12, т.е. 2 пъти:

Факториела е експоненциална функция, която нараства изключително бързо. В Таблица 1 е показано как броят на възможните обиколки расте с увеличаване на броя на бензиностанциите.

|  |  |
| --- | --- |
| Брой на бензиностанциите | Брой на възможните обиколки |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 10 |  |
| 15 |  |
| 18 |  |
| 20 |  |

Таблица 1.

Нарастване на възможните обиколки според броя на бензиностанциите

От Таблица 1 виждаме, че ако получим като входни данни 18 бензиностанции (17 + началната), което е максимумът по условие, ще имаме възможни обиколки! По време на писането на този документ, моят компютър успява да направи изчисление за 1000800 обиколки за 1 сек. Това означава, че **(18-1)! ще отнеме около** **11 години**. Увеличаването със само още 3 бензиностанции т.е **20! ще отнеме 77939 години**.

Въпреки че в условието на задачата не е дадено крайно време за намиране на резултат, можем да приемем, че ако искаме да получим дипломата си от УНИБИТ по-рано от 11 години след предаването на решението, ще трябва да намерим по-ефективен начин за справяне с проблема.

# Алгоритъм за Приблизително Решение