Design & Analysis of Algorithms

A Project Report

Submitted by

Kashish Madan (179202069)

Krishna Rakesh Goyal (179301101)

Mridu Kejariwal (179301121)

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

At

MANIPAL UNIVERSITY
JAIPUR

MANIPAL UNIVERSITY JAIPUR

April 2019

DECLARATION BY THE CANDIDATE

I hereby declare that the work, which is going to be presented in the Project, entitled **"Car Fuelling : Travelling through cities with minimum stops"** to be submitted in partial fulfilment of the requirement for the award of the degree of "Bachelors of Technology" in Department of CSE with Specialization in DAA and to be submitted to the Department of CSE, Manipal University Jaipur is an authentic record of my own investigations carried out under the supervision of Mr. Tarun Jain , Assistant Professor, SCIT, Manipal University Jaipur.

I have not submitted the matter presented in this project anywhere for the award of any other degree.

Student Name- Kashish Madan (179202096)
Krishna Rakesh Goyal (179301101)
Mridu Kejariwal (179301121)

ACKNOWLEDGEMENT

The enduring pages of the work are the cumulative sequence of extensive guidance and arduous work. I wish to acknowledge and express my personal gratitude to all those without whom this work could not have been reality.

I feel very delighted to get this rare opportunity to show my profound senses of reverences and indebtedness to my esteemed teacher, Mr. Tarun Jain for his keen and sustained interest, valuable advice, throughout the course of which led my report, to a successful completion. For this kind act of consideration, I beholder to him in special manner and no one can fully convey my feelings of respect and regard for them.

TABLE OF CONTENTS

ABSTRACT

Greedy Algorithm is a search technique used in computing to find the optimal solution to a computational problem that minimizes a function. Greedy Algorithm is used to solve the Car Fuelling Problem where one must find the minimum number of cities to selected to refuel the gas tank and then reach the destination. Greedy Algorithm is a computer algorithm that search for good solutions to a problem from various possible solutions. The basic elements of Greedy Algorithm are chromosomes, fitness function, selection, crossover and mutation. The Car Fuelling Problem is well known in the field of combinatorial optimization.

Since it is an NP-complete problem, there is no efficient method to solve this problem and give the best result. Some algorithms give optimal solution, but some other algorithms give the nearest optimal solution.

The greedy algorithm is a heuristic method which is used to improve the solution space for this problem. The greedy algorithm results in nearest optimal solution within a reasonable time. This project mainly focuses on the comparative study of different selection methods and crossover operators in greedy algorithm to solve Car Fuelling Problem and compute the results.

INTRODUCTION

Background

Suppose a person is making a travel plan driving from city 1 to city n, where n>1. Following a route that will go through cities 2 through n-1 in between. The person knows the mileages between two adjacent cities and knows how many miles a full tank of gasoline can travel. Based on this information, the problem is to minimize the number of stops for filling up the gas tank, assuming there is exactly one gas station in each of the cities.

Project Motivation

The motivation for doing this project was primarily an interest in undertaking a challenging problem related to algorithms. The opportunity to learn something new by solving a problem was appealing. Also, the problem focuses on saving user's time by efficiently planning the cities where the user needs to refuel the tank.

Aim and Objective

The aim of the project is to solve the Car Fuelling Problem with minimum number of stops using the Greedy Algorithm.

The following points present the objective:

- To study the Greedy algorithm.
- To study various parameters and operations of Greedy algorithm.
- To find the solution for the "**Car Fuelling - Travelling through cities with minimum stops"** problem using the Greedy algorithm.
- To make a comparative study of different selection methods and crossover operators and finally report the results.

IMPLEMENTATION

Technology Used

The logic for Car Fuelling problem can be formed by using-

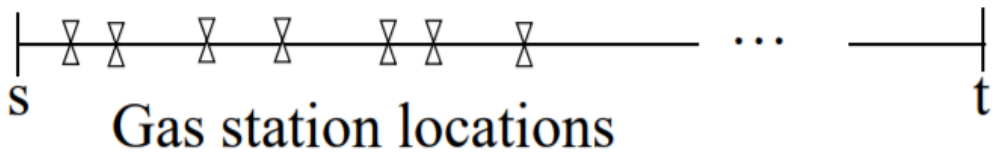
- **Greedy Approach**

Among all the algorithmic approaches, the simplest and the most straightforward approach is the Greedy method. In this approach, the decision is taken based on current available information without worrying about the effect of the current decision in future.

Greedy algorithms build a solution part by part, choosing the next part in such a way, that it gives an immediate benefit. This approach never reconsiders the choices taken previously. This approach is mainly used to solve optimization problems. Greedy method is easy to implement and quite efficient in most of the cases. Hence, we can say that Greedy algorithm is an algorithmic paradigm based on heuristic that follows local optimal choice at each step with the hope of finding global optimal solution.

In many problems, it does not produce an optimal solution though it gives an approximate (near optimal) solution in a reasonable time.

Greedy idea: Go as far as you can go before stopping for gas!



Algorithms Used

Algorithm:

find_gas_stops():

current position = start position;

while (current position < end position)

compute the position at which car will run out of gas.

if (that position < end position)

then find closest gas station before reaching that position.

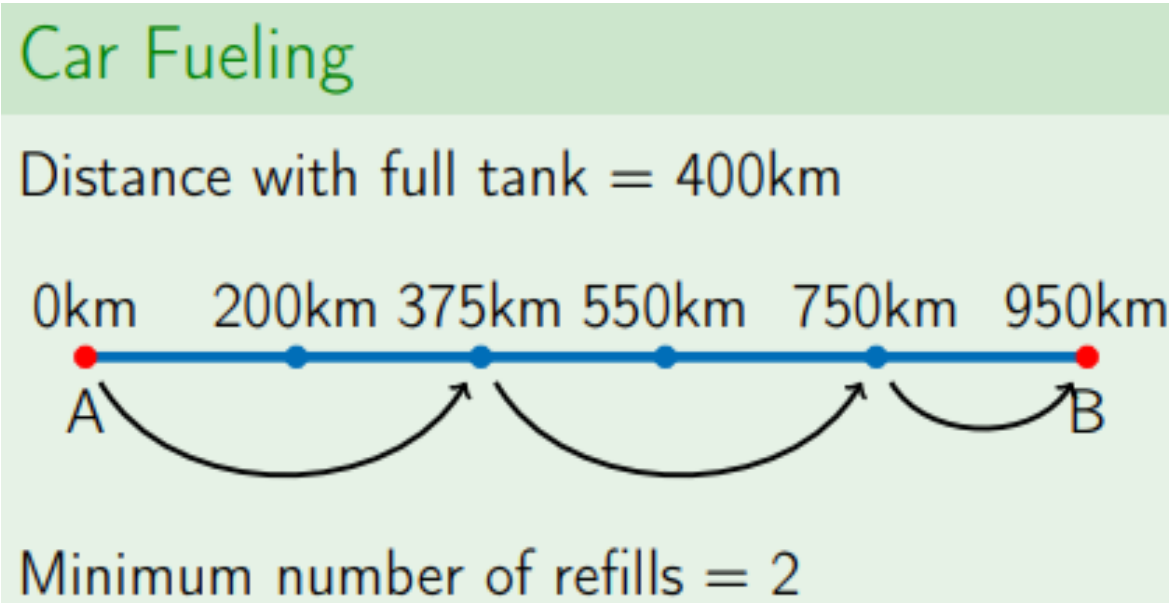
output fill up gas at that gas station.

current position = that gas station location

else

current position = end position;

/ reached */*



Explanation

Input: A car which can travel at most L kms with full tank, a source point A, a destination point B and n gas station at distances x1, x2, x3.. xn in kms from A along the path from A to B.

Output: The minimum number of refills to get from A to B, besides refill at A.

This program has loop nested within another loop. So it seems to have O(n*n) run-time. But, the run time of the program is O(n) as

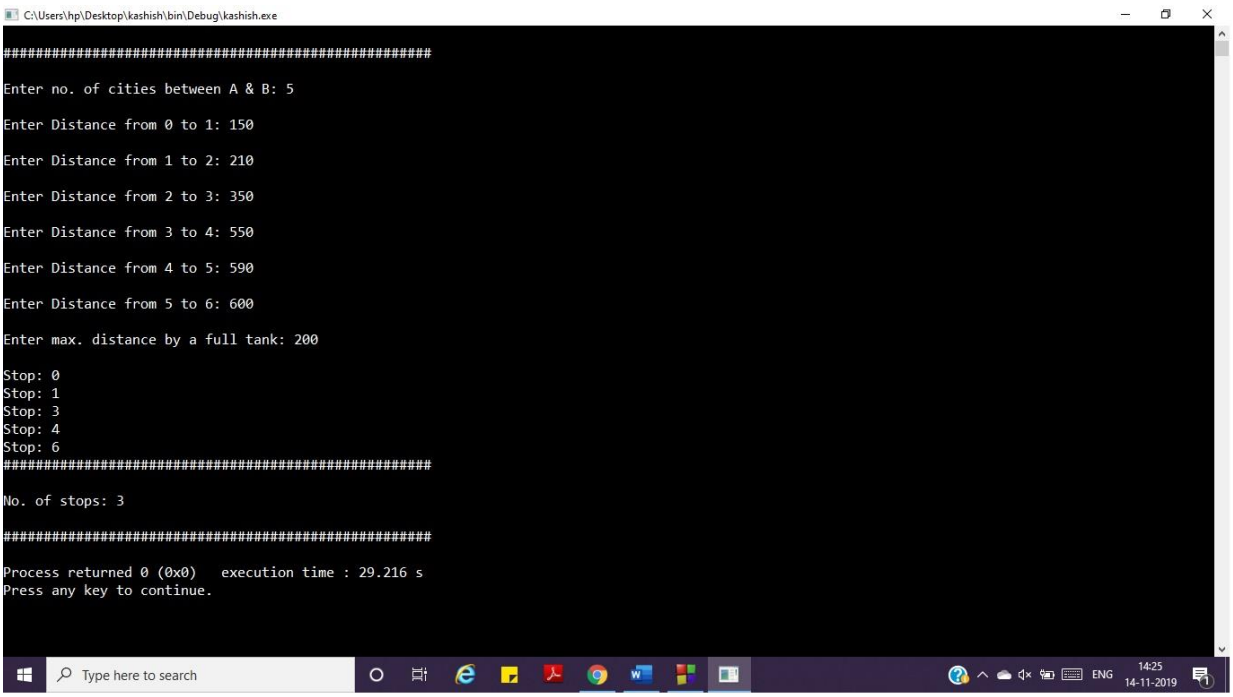
- CurrentRefills can be at most n-1
- numRefills can be at most n
- So, there will be at most n+1 operations
- O(n+1) => O(n)

SOURCE CODE AND OUTPUT

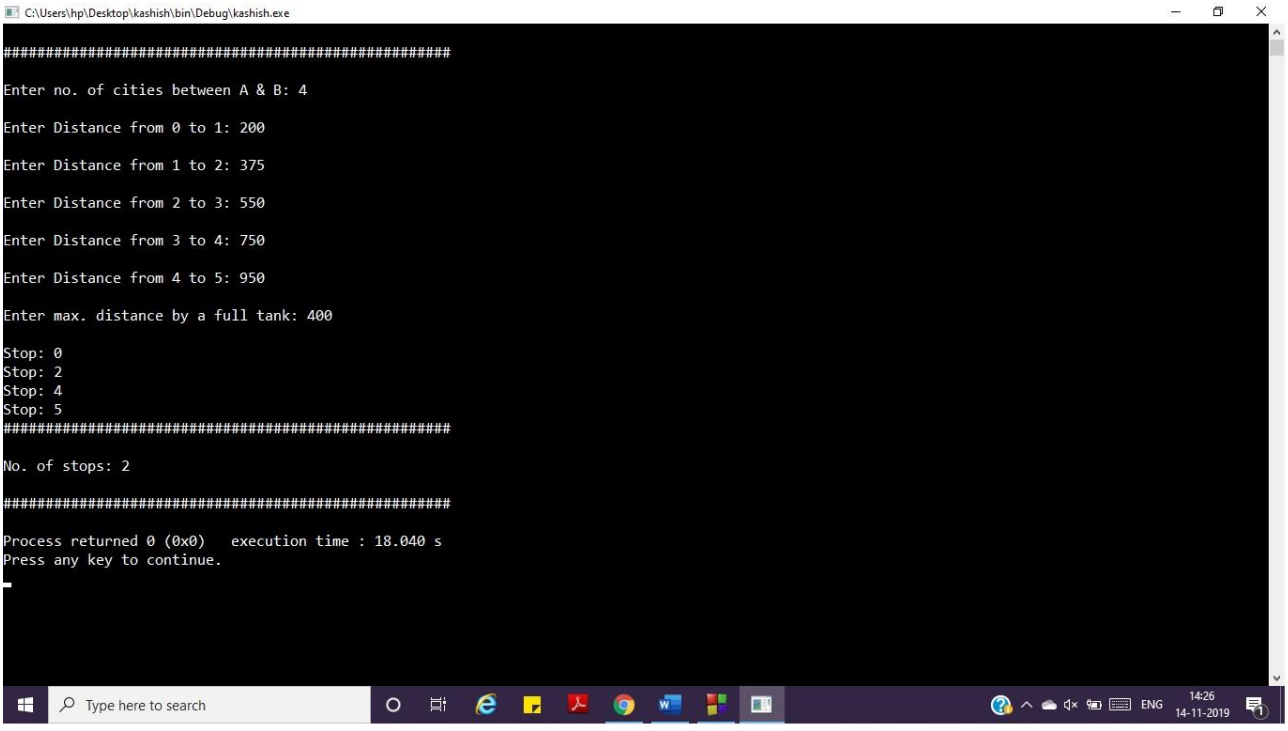
```
int MinRefills(int x[],int n,int L)
{
    int count=0,curr=0,last=0;
    printf("\nStop: %d",curr);
    while(curr<(n-1))
    {
        last=curr;
        while(curr<(n-1))
        {
            if((x[curr+1]-x[last])<=L)
                curr+=1;
            else break;
        }

        if(curr==last)
            return -1;
        if(curr<(n-1))
        {
            printf("\nStop: %d",curr);
            count++;
        }
    }
    printf("\nStop: %d",curr);
    return count;
}


void main()
{
    int i,n,L,count=0;

printf("\n##############################################
########\n");
    printf("\nEnter no. of cities between A & B: ");
    scanf("%d",&n);
    n=n+2;
    int x[n];
```

```
    x[0]=0;
    for(i=1;i<n;i++)
    {
        printf("\nEnter Distance from %d to %d: ",i-
1,i);
        scanf("%d",&x[i]);
    }
    printf("\nEnter max. distance by a full tank: ");
    scanf("%d",&L);
    count=MinRefills(x,n,L);

printf("\n##############################################
########\n");
    if(count==-1)
        printf("IMPOSSIBLE");
    else
        printf("\nNo. of stops: %d \n",count);

printf("\n##############################################
########\n");
}
```

Output 1



Output 2



CONCLUSION

After studying all the techniques to solve this problem, it is concluded that the traditional algorithms have major shortcomings: firstly, they are not suitable for negative edge networks; secondly, they exhibit high computational complexity.

Although, researchers have got remarkable success in designing a better algorithm in term of space and time complexity to solve shortest path problem. But greedy approach for Car Fuelling Problem is efficient and can solve the problem in short time. The performance of this algorithm depends on the appropriate setting of parameters. These parameters depend on the problem instance in hand and on the required solution accuracy.

REFERENCES

- https://www.coursera.org/lecture/algorithmic-toolbox/car-fueling-implementation-and-analysis-shwg1
- https://www.geeksforgeeks.org/greedy-algorithms/
- https://brilliant.org/wiki/greedy-algorithm/
- https://medium.com/cracking-the-data-science-interview/greedy-algorithm-and-dynamic-programming-a8c019928405
- https://medium.com/@TheGeekiestOne/greedy-algorithms-the-car-fueling-problem-a35ccc9bb011