

VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



DISCRETE STRUCTURE

Assignment 1

Using Dynamic Programming
To Solve The Travelling Saleman Problem

Instructors: Nguyen Van Minh Man, *Mahidol University*
Tran Tuan Anh, *CSE-HCMUT*

Author: Truong Gia Ky Nam

ID: 2352787

Email: nam.truonggiaky@hcmut.edu.vn

Ho Chi Minh City, May 2024

Contents

1 Introduction 1

2 The Traveling Salemen Problem 1

2.1 History 1

2.2 Approaching 1

2.3 tspprogram 1

2.4 Travling function 1

2.5 2

2.6 2

2.7 2

2.8 Dynamic Programming 2

3 Conclusion 2

Abstract

This document was made during the class CC03 from course Discrete Structure semester 232 with the instruction from Dr. Nguyen Van Minh Man and Dr. Tran Tuan Anh. This document serves as the assignment report for the Assignment 1 that Dr. Minh Man and Dr. Tuan Anh gave us during the course. By gather information in searching information on the Internet, our group is able to gather required information about the topic that is given and summarized in this report.

1 Introduction

In this report, I will show how we can use the dynamic programming technique to solve the Traveling Saleman Problem with clearly explanation about the approaching method and the purpose of each functions of the source code.

2 The Traveling Salemen Problem

2.1 History

2.2 Approaching

2.3 tspprogram

```
1 void tspprogram(){
2     int cityMap[numberOfVertices][numberOfVertices];
3     for (int i = 0; i < numberOfVertices; i++){
4         for (int j = 0; j < numberOfVertices; j++){
5             if (i == j){
6                 cityMap[i][j] = 0;
7             }else{
8                 cout << "Input the distance from city " << char(i + 65) << " to city " << char(j + 65) << ": ";
9                 cin >> cityMap[i][j];
10            }
11        }
12    }
13
14    char startCity;
15    cout << "Input the start city: ";
16    cin >> startCity;
17
18    Traveling(cityMap,startCity);
19 }
```

2.4 Travling function

```
1 void Traveling(int cityMap[][numberOfVertices],char startVertex){
2     vector<vector<int>> memo(numberOfVertices,vector<int>(1 << numberOfVertices,0));
3     int source = int(startVertex - 'A');
4
5     setUp(cityMap,memo,source,numberOfVertices);
6     solve(cityMap,memo,source,numberOfVertices);
7
8     cout << findOptimalTour(cityMap,memo,source,numberOfVertices);
9 }
```



2.5

2.6

2.7

2.8 Dynamic Programming

3 Conclusion

BLA BLA BLA



```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6      //Use this to print Hello World
7      cout << "Hello World";
8      return 0;
9  }
```

References

- [1] CVX Introduction “**link:** <http://cvxr.com/cvx/doc/intro.html/>”, *What is CVX*.