DOKUZ EYLUL UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING

CME2204 – Algorithm Analysis

ASSIGNMENT – 2

Dynamic Programming and Greedy Approach

by
Nilay YÜCEL
2017510082

May, 2020
İZMİR

CONTENTS

CHAPTER ONE

INTRODUCTION

This assignment is the second assignment within the scope of CME2204- Algorithm Analysis course. Assignment was given on 17.04.2020 and is expected to be delivered on 28.05.2020. The total duration of the assignment is about 5 weeks. The expected assignment is to obtain detailed information about the "Dynamic Programming" and "Greedy Approach". There are two main problems with Assignment. We have to solve these problems with different methods.

Part-1

Assume that you are the owner of a car company. Your company has enough employees to produce **'p'** cars for each month. However, the number of the demand for the cars differs from month to month. You should design a sales plan for the next ***'x'*** months. Consider ***'i'*** is the index of each month ($i=1,\ldots,x$) and m_i is the demand for i^{th} month. If your company needs to produce more than ***'p'*** cars for a month, you can hire some interns, paying ***'d'*** TL per car for that month. Moreover, if your company keeps any unsold car at the end the month, you should pay a **'garage cost'.** The garage cost will be calculated by the function *G(j)*, where,

The first scenario is the one above. We are expected to produce a solution that will reduce the monthly expense of a car company. The company must pay garage money for every car it stores. The company should also pay a salary for every intern it hires. We are expected to ensure that the company chooses the best scenario based on monthly demand. We are expected to find the result with two different methods.

Part-2

Besides, you must invest the payment that earned from your sales. Cost of each car is **'B'** TL and you get half of the price at the beginning of the month and the rest will be taken at the end of the month. You have offers from **'c'** different investment companies. Each investment company offers different rate for each month. At the end of each month, you can change your investment company by paying a taxes at a rate **'t'** of your invested money or continue with the same investment company without paying any taxes.

The second scenario is the one above. We are expected to make the most profit by investing the money we receive in various investments. However, the rate of these investmants changes every month. At the same time, when it is desired to switch between investments, there is an obligation to pay taxes. Considering these possibilities, we should get the most profit with two different algorithms.

In this report, the algorithms used for the assignment given the problems encountered and their solutions, etc. contains information.

CHAPTER TWO

EXPLANATION OF ALGORITHMS

Part-1 Dynamic Programming:

Firstly, an array was created to keep monthly expenses. The number of lines of this array was determined by month **(x)**. Its columns were determined according to the **total number of demands (∑i)**. The number of columns was determined according to the total demand because the maximum number of cars it can hold in its garage every month is necessary for us to make the calculations correctly. As we walk through Array, we need another extra for loop that runs between our columns. This for loop shows us how many cars we kept in the previous month and the money paid to the garage previous month. As a result of this for loop, we get many values, but we keep the best. Then we complete the related line according to the extra cars production in the current month. If we do not have enough cars and the demand is more than production **(p)**, we do not add intern payments (**d**) to our total expense by hiring one intern per car. At the same time, if we have produced more cars, the garage fee for storage is also added to the total expense. As a result of all these processes, we calculate our total minimum expense.

- Run Time Complexity: O(x*∑i*∑i)
- Space Complexity: O(x*∑**i)**

Part-1 Greedy Approach:

The "Greedy Algorithm" doesn't give us the most accurate result, but it gives the best result it can find instantly. Firstly, an array was created that can hold monthly expenses. The size of this array was determined by the number of months**(x)**. A for loop was created so that we could take into account the number of cars in the garage for the first month. This for loop rotates in an array created according to the **total demand(∑i)** and contains garage prices. This was not mandatory, but this possibility is also taken into account in my algorithm. The best result is recorded after the expenses of the first month are determined by the number of cars stored in the garage the previous month. Since the “Greedy Algorithm” will find the best result instantly, the extra vehicle generation status is ignored.

At the same time, if the demand is less than production, cars are produced according to demand, because paying extra storage is not appropriate due to the nature of our algorithm. If the demand is more than production **(p)**, extra interns are taken and intern's payments **(d)** are added to the total expense. As a result of all these transactions, we get the minimum expense instantly.

- Run Time Complexity: O(x*∑i) // According to my algorithm. May be O(x)
- Space Complexity: O(x**)**

Part-2 Dynamic Programming:

Firstly, how much money the company will have per month according to the sales cost **(B)** of each car is found and then this income is recorded in an array by making the necessary transactions. Secondly, a another array was created. Array's number of lines was determined by month **(x)**. The number of columns was determined by the number of investments **(c)**. As we walk through Array, we need another extra for loop that runs between our columns. This array is used to make the transition according to investment in the previous month. If you want to switch from a different investment, the tax fee should be deducted from the income. Looking at all the revenues of the past month, the best possibility is recorded, also taking into account the tax **(t)** and interest rates. As a result of all these processes, we calculate our total maximum profit.

- Run Time Complexity: O(x*c*c)
- Space Complexity: O(x*c)

Part-2 Greedy Approach:

The "Greedy Algorithm" doesn't give us the most accurate result, but it gives the best result it can find instantly. Firstly, how much money the company will have per month according to the sales cost **(B)** of each car is found and then this income is recorded in an array by making the necessary transactions. Secondly, a another array was created. Array's number of lines was determined by month **(x)**. The number of columns was determined by the number of investments **(c)**. As you move through one line of Array, the best possibility in that line is recorded. Calculations are made according to this possibility, which is recorded in the next month. If the bank selected in the previous month and the bank in the

current month are different, the tax **(t)** is deducted from income. Then the best possibility in the current line is recorded. In this way, the best profit is obtained instantly.

- Run Time Complexity: O(x*c)
- Space Complexity: O(x**)**

CHAPTER THREE

PROGRESS SUMMARY

3.1. Completed Tasks

Everything requested in Assignment has been completed. According to the related scenarios, the best results were obtained with “Dynamic Programming” and “Greedy Approach”.

3.2. Incomplete Tasks

There are no parts that are not completed.

CHAPTER FOUR

PROBLEMS ENCOUNTERED

Only in Part-1, a problem was encountered in the Greedy approach. Since the Greedy approach would give the best results instantly, it was difficult to register a car in the garage if it was suitable for this approach. For this reason, the algorithm was changed too much and too many wrong results were obtained. But most recently, an algorithm has been developed for the number of cars available in the garage only for the first month. The possibility of storing carsin the garage of the other months was ignored and the best solution was reached instantly.

CHAPTER FIVE

OVERALL ASSESSMENT OF THE PROJECT

In this assignment, "Dynamic Programming" and "Greedy Approach" were well analyzed and the necessary comparisons were made according to the given scenarios.

In this report, all the information about the assignment is given in detail.

CHAPTER SIX

REFERENCES

- https://www.geeksforgeeks.org/dynamic-programming/
- https://www.geeksforgeeks.org/greedy-algorithms/