

DOKUZ EYLUL UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING

CME2204 ALGORITHM ANALYSIS
ASSIGNMENT REPORT

DYNAMIC PROGRAMMING AND GREEDY APPROACH

by
Rıdvan Özdemir
2017510086

Lecturers
Dr. Zerrin Işık
Ali Cüvitoğlu
Ezgi Demir

IZMIR
28.05.2020

CHAPTER ONE

PROGRESS DESCRIPTION

The aim of the project is to develop a dynamic programming and greedy approach that minimizes the production costs and maximizes the profit from investments at the end of the ' x ' months.

This project consists of two parts. There are a car company in both parts. According to first part, this company has enough employees to produce ' p ' cars for each month. However, this company has to satisfy the number of the demand for the cars for each month. Hence, company can hire some interns, paying ' d ' TL for per car, if it does not satisfy the demand or keep any unsold car at the end the month, paying '**garage cost**'. Consequently, The project must minimize the cost for this part by using dynamic programming and greedy approach.

In the other hands, the second part of the project wants us to maximize the profit that is coming from investment companies at the end of the ' x ' months by using dynamic programming and greedy approach. 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. Also, The company has offers from ' c ' different investment companies. Each investment company offers different rate for each month. If the company decide to change the investment company at the end of the month, it must pay a taxes at a rate ' t '. Program must maximize the profit under this conditions.

CHAPTER TWO

TASK SUMMARY

2.1. Completed Tasks

Text file that has demand for each month is read and stored into a special "months_demands" array.

Text file that has investment rates of each companies for each month is read and stored into a special "investments" array.

Text file that has garage costs for each month is read and stored into a special "garage_costs" array.

A function that calculates the profit for given ' x ' months is implemented by using dynamic programming approach.

A function that calculates the cost of the company for given ' x ' month is implemented by using dynamic programming approach.

A function that calculates the profit for given 'x' months is implemented by using greedy approach.

A function that calculates the cost of the company for given 'x' month is implemented by using greedy approach.

2.1. Incomplete Tasks

There is no incompleted task.

2.1. Additional Improvements

No additional improvement is expected for the project.

CHAPTER THREE

EXPLANATION OF ALGORITHMS

3.1. Algorithm and Solution Strategies

There are four different approaches and solution strategies because each part of the assignment needs to special approach to be solved. First of these parts is calculating profit at the end of 'x' month by using dynamic programming approach. In this part, I used 2 matrices, one of them is size of $x \cdot c$ and its name is 'profits'. It stores the profit that is coming from the each company for each month. Other matrix is size of x and its name is 'companies'. It stores the companies that are selected for months. This matrices indices like $i=0,1,\dots$, are used for the months.

c1
----	------	------	-----

"companies" array for $x = 4$

c1	c2	c3	c4
504	485	470	494
...

"profits" array for $x=3$ and $c=4$

I have a main loop for this part of assignment. This loop consists of 3 nested loop. First of them iterates until 'x', second and third of them iterate until 'c' as company count. At the outside of this main loop, I calculated profit and the selected company for the first month. For the others month, I calculated the other months by using based on the companies I kept in the array and using the profits array inside of the main loop. This two arrays are reducing the number of loop that I have to do to find the maximum profit of month.

Second of these parts is calculating the cost at the end of 'x' month by using dynamic programming approach. Firstly, I calculated total demand at the end of 'x' month and stored it into a variable is called 'R'. In this part, I used a matrix that is size of x.R and its name is 'costs'. It stores the cost for each month by based on garage costs and hired interns cost.

	0	1	2	R
0							
1							
2							

"costs" array for x=2 and value of R

This matrix indices like $i=0,1,\dots,R$, are used for the car numbers of stored in the garage. I took zeroth month for this part, because there may be remaining cars from zeroth month, in this situation I have to pay garage cost. I have a main loop for this part of assignment. This loop consists of 3 nested loop. First of them iterates until 'x', second and third of them iterate until 'R' as total demands. At the outside of this main loop, I initialized the "costs" array for the zeroth month by using the "garage_costs array". For the others month, I calculated costs using based on this "costs" array.

Third of these parts is calculating profit at the end of 'x' month by using greedy approach. I just calculated the profit that is coming each company for current month and stored its maximum value into a variable. Also, I stored the selected company into a variable. I don't use any array because company selection of the current month does not effect the previous months. I have a main loop for this part of assignment. This loop consists of 2 nested loop. First of them iterates until 'x', second of them iterate until 'c' as company count. At the outside of this main loop, I calculated profit and the selected company for the first month. For the others month, I calculated the other months by using these two variables. If company of current month and company of the previous month are different, tax will be taken.

Last of these parts is calculating the cost at the end of 'x' month by using greedy approach. I just calculated the cost regardless of the remaining cars of previous months. I just calculated the cost of the hired interns. I did not put cars into garage thus, I did not pay garage cost.

CHAPTER FOUR

PROBLEMS ENCOUNTERED

I did not encounter a technical problem during the coding. My problem was about the creating the greedy cost solution strategy. I have considered about that should I have not consider the zeroth month and remaining cars of previous months. Also, should I have not consider the event that put cars into garage ? Because if I consider these situations, it becomes kind of dynamic programming. That's why, I calculated the cost regardless of these situations.

CHAPTER FIVE

TIME AND SPACE COMPLEXITY

DP-PROFIT: In this part, I have nested loop that I explained in algorithms and solution strategies. This nested loop consist of 3 parts. First of them iterates until 'x' as count of month. Second and third of them iterates until the 'c' as company number

Time Complexity = (month count).(company count).(company count)

Time Complexity = $x.c.c = O(x.c.c)$

I have two matrices in this part. One of them is "profits" array that has size of $x.c$. Other one is "company array that has size of x .

Space Complexity = (month count).(company count)

Space Complexity = $x.c = O(x.c)$

DP-COST: In this part, I have nested loop that I explained in algorithms and solution strategies. This nested loop consist of 3 parts. First of them iterates until 'x' as count of month. Second and third of them iterates until the 'R' as total demands.

$$\text{Time Complexity} = (\text{month count}).(\text{total demands}).(\text{total demands})$$

$$\text{Time Complexity} = x.R.R = O(x.R.R)$$

I have one matrix in this part. It is "costs" array that has size of (x+1).R. Also, I stored the zeroth month in this array thus, it is (x+1).

$$\text{Space Complexity} = (\text{month count}+1).(\text{total demands})$$

$$\text{Space Complexity} = (x+1).R = O((x+1).R)$$

GREEDY-PROFIT: In this part, I have nested loop that I explained in algorithms and solution strategies. This nested loop consist of 2 parts. First of them iterates until 'x' as count of month. Second of them iterates until the 'c' as company number

$$\text{Time Complexity} = (\text{month count}).(\text{company count})$$

$$\text{Time Complexity} = x.c = O(x.c)$$

I do not have any matrices in this part. I just use some variables. These constant sized variables do not allocate any **extra** space. Which means this loop clearly has constant space complexity **O(1)**.

$$\text{Space Complexity} = O(1)$$

GREEDY-COST: In this part, I have just one loop that I explained in algorithms and solution strategies. This loop iterates until 'x' as count of month.

$$\text{Time Complexity} = (\text{month count})$$

$$\text{Time Complexity} = x = O(x)$$

I do not have any matrices in this part. I just use some variables. These constant sized variables do not allocate any **extra** space. Which means this loop clearly has constant space complexity **O(1)**.

$$\text{Space Complexity} = O(1)$$

CHAPTER FIVE

CONCLUSION

I completed all expected task of this assignment. I've learned dynamic programming and greedy approach and why we should use them on some problems. In this assignment, if we did not use them, reaching the solution will take much more time. For example, If I used the brute-force in this assignment, time complexity will be c^x in this assignment but time complexity is $x.c.c$ by using dynamic programming approach.

REFERENCES

<https://www.geeksforgeeks.org/greedy-algorithms/>

<http://bilgisayarkavramlari.sadievrenseker.com/2008/03/24/acgozlu-yaklasimi-greedy-approach/>

<https://www.geeksforgeeks.org/dynamic-programming/>

https://www.tutorialspoint.com/design_and_analysis_of_algorithms/design_and_analysis_of_algorithms_longest_common_subsequence.htm

DESIGN AND ANALYSIS OF ALGORITHMS, Herbert Edelsbrunner, Zhiqiang Gu, 2008, page. 11- 16