FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT
Tınaztepe Yerleşkesi, Buca-Kaynaklar, Dokuz Eylül Üniversitesi, İZMİR, TÜRKİYE
14.05.2023
Report: CME 2204 Algorithm Analysis Assignment II
Prepared by: Yusuf Gassaloğlu (2020510034)

- **Introduction**: Understanding the basics of dynamic programming and to be able to use that on the problem. Finally find the solution.

- **Project Description**: Assume that you are the owner of a football club. Your club has enough trainers to improve and promote ‘p’ players to the main squad from the youth team. Some other clubs ask to rent players from your club and make several demands for renting players from your club. The number of demands can be different from year to year. You should design a renting plan for the next ‘n’ years. Consider ‘i’ is the index of each year (i=1,…,n) and i is the number of players will be requested to rent for i.th year. If your club needs to promote more than ‘p’ players for a year, you can hire some coaches, paying ‘c’ TL costs per player for that year. However, if your club keeps any unrented player in that year, you should pay a ‘salary’ for a such player.

- **Problems Encountered**: Time was main problem. I had to do the assignment at a busy time.

- **What is dynamic programming?**

Dynamic Programming is mainly an optimization over plain recursion. Wherever we see a recursive solution that has repeated calls for same inputs, we can optimize it using Dynamic Programming. The idea is to simply store the results of subproblems, so that we do not have to re-compute them when needed later. This simple optimization reduces time complexities from exponential to polynomial.[1]

- **How Dynamic Programming Was Used in This Assignment?**

Knapsack algorithm is used in this assignment. What is knapsack algorithm? The knapsack problem is an optimization problem used to illustrate both problem and solution. It derives its name from a scenario where one is constrained in the number of items that can be placed inside a fixed-size knapsack. Given a set of items with specific weights and values, the aim is to get as much value into the knapsack as possible given the weight constraint of the knapsack.[2] But in this assignment knapsack minimizes the cost.

We got 2D matrix. Row length is total demand value + 1. Column length is equal to year + 1First row of matrix shows salaries. First column of matrix shows years. The algorithm walks every cells one by one and calculates every possibilities by looking past years salaries. First store all possibilities for a cell in an array list and find minimum fill cell.

After calculations the time complexity of the algorithm is O(n^3) (3 nested loops). Space complexity is O(1).

- **References**

[1] https://www.geeksforgeeks.org/dynamic-programming/
[2] https://www.techopedia.com/definition/20272/knapsack-problem