



Project Description:

The aim of this project is to calculate a rental plan and additional costs such as hiring a coach or paying salaries, if necessary, in a scenario where the number of players to be promoted from the youth team of a football club to the A team is known and other clubs have requests to hire players from this club.

The project aims to determine the number of players (y_i) that will be requested each year for the club's next 'n' year period and to plan how the club will respond to these requests. If the club has to promote more than 'p' players to the first team in a year, additional coaches can be hired to support the development of these players. However, if the club does not hire any players, it will have to pay their salaries. The project aims to help the club make the most economical and efficient decisions in this scenario.

Completed – Uncompleted Tasks:

All tasks of the project have been completed.

The salary and demand txt files received in the project were scanned and the two were assigned to an array separately. Then the array holding the demands was scanned and a code was written that selects the most efficient possibility for two years by looking at the desired demands. The code looked at the demand in year n and the annual production, p. If the demand was higher than the production, it was seen that extra coaches should be hired and this was printed on the screen. If, on the contrary, the demand is less, it looked at the following year and tested whether it makes more sense to pay the salaries of the required number of players in the following year or to produce the required number of players in the following year and calculated how many players to save for the following year. In this way, the greedy algorithm was completed in a way that provided the customer with the options that made the most sense for two years.

Run-time Complexity and Space Complexity:

O(n) for the “ga” method is as follows:

$$O(n \cdot p) = O(N \cdot p)$$

The algorithm runs n times for n years, and when checking each year, it runs another for the total number p of years in that maximum year in order to include the next year in the calculation. Therefore it is calculated as O(N.P).

Space complexity for the dp method is as follows:

$$O(1).$$

Since the program does not use any array, it is independent of n and the value is O(1).

Dynamic Programming and Greedy Approach Comparison:

The main differences between the greedy approach and dynamic programming are that the greedy approach seeks the best in the moment, while dynamic programming aims to find the most sensible option at the end of all time. Therefore, the greedy approach cannot guarantee the maximum possible efficiency, while dynamic programming aims to find the best among all possibilities. Another difference is that in the greedy approach, the problem is considered at that moment without thinking about the overall problem, while dynamic programming calculates all the possibilities that can be done at that moment and stores them. In the later stages, it thinks about what it has stored and actually divides the problem into parts and looks for the perfect option among them.

In general, the greedy approach can work well for small problems, but if we have big data or a long problem, dynamic programming will save us a lot of time and efficiency.