Greedy Approach

Introduction

The problem at hand is to design a renting plan for a football club that receives requests from other clubs to rent players from their youth team. The plan should cover a period of 'n' years, during which the club can promote a limited number of players, denoted by 'p', at no cost. If the number of players requested for a given year, denoted by 'demand[i]', exceeds 'p', the club can choose to hire additional trainers at a cost of 'c' to promote the remaining players. Furthermore, if the club has any unrented players each year, they will need to pay a 'salary' for each of those players, denoted by 'salary[i]'. To solve this problem, the program uses greedy approach principles.

Algorithm

The program has an iterative function called Greedy. The function takes several parameters, including the period, the number of promotable players, the number of players to hire each year, the cost of hiring players, and the cost of reserving players. The algorithm first calculates the difference between the player demand and the number of promotable players plus players at hand. If the difference is greater than or equal to zero, we rent players according to the difference. This is always the optimal way. But if the difference is less than zero, we first need to check the next year's demand. If the demand is positive, there is no cost. Otherwise, we take the smaller value between this year's difference and the next year's difference. This value represents the number of players we need to hire.

Time Complexity and Space Complexity

There is no subproblem to iterate over in this algorithm. This is because the algorithm chooses the local optimal solution in every step without iterating over subproblems. Therefore, this algorithm has a time complexity of O(n), as it finishes with a single iteration.

The algorithm does not dynamically create new variables or dynamically allocate new memory. It only updates current variables, and this makes the space complexity O(1).