HOMEWORK 5

Yunus Emre Geyik / 1801042635

All algorithms are also commented on the code line by line.

Question – 1

(a)

Dynamic programming is technique for solving problems with overlapping subproblems. Rather than solving overlapping again and again, solve each of the smaller subproblems only once. Via bottom to up method.

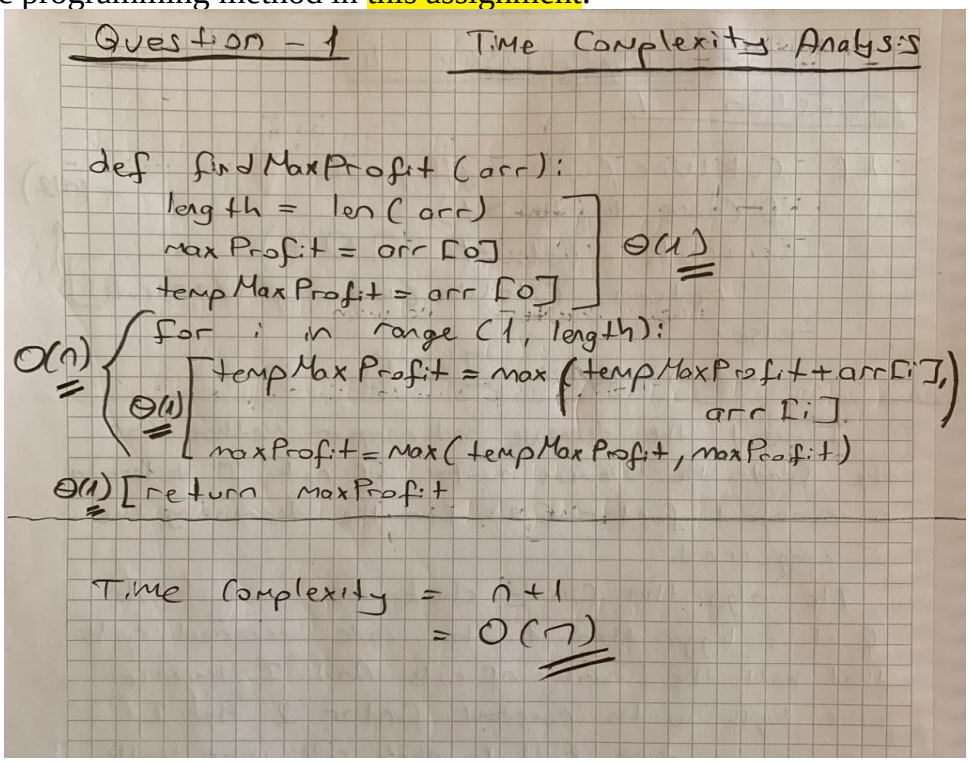
The recursive relation of this algorithm is : tempMaxProfit = max(tempMaxProfit + arr[i], arr[i]). By iterating through the array, we add the current array element (i'th index value) to the value of the tempMaxProfit variable, and with the result we get, we assign the tempMaxProfit variable, whichever is greater than the current element. Thus, we solve the sub-problems we have obtained to find the max profit only once.

In the algorithm, we initially set the variables maxProfit and tempMaxProfit as the first element of the array. Then we do the following in a loop that start from 1 index to the last index:

By iterating through the array, we add the current array element to the value of the tempMaxProfit variable, and with the result we get, we assign the tempMaxProfit variable, whichever is greater than the current element. If tempMaxProfit is greater than maxProfit, update maxProfit equals to tempMaxProfit. And the function returns the resulting maxProfit result.

(b)

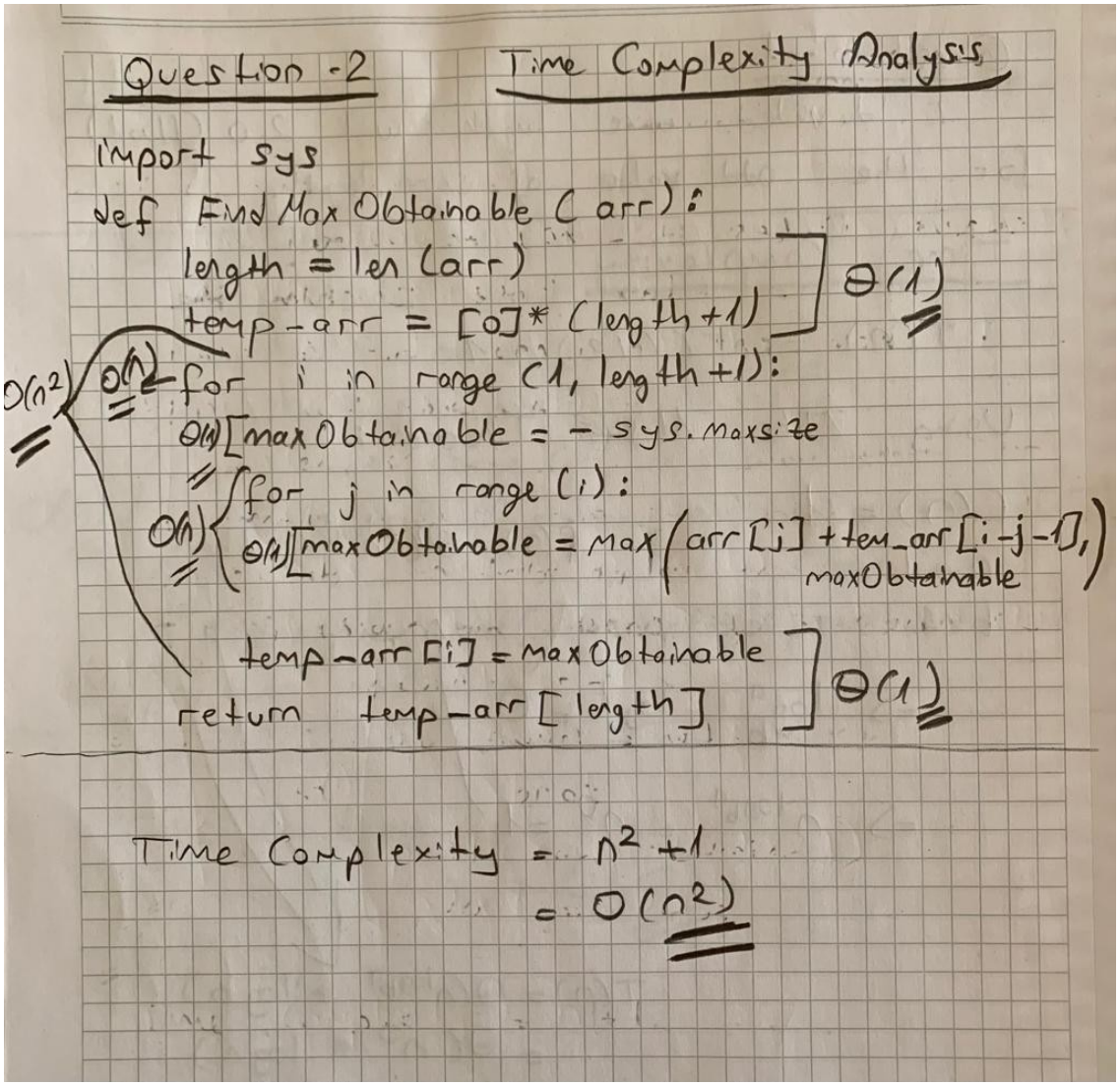
I solved this problem with divide and conquer method in a previous assignment and the time complexity result I got was O(n.log(n)) . But in this assignment, I solved it with the dynamic programming method and the time complexity result I got is O(n). When we compare these two algorithms, it will be more effective to use the algorithm I designed with the dynamic programming method in this assignment.



Question – 2

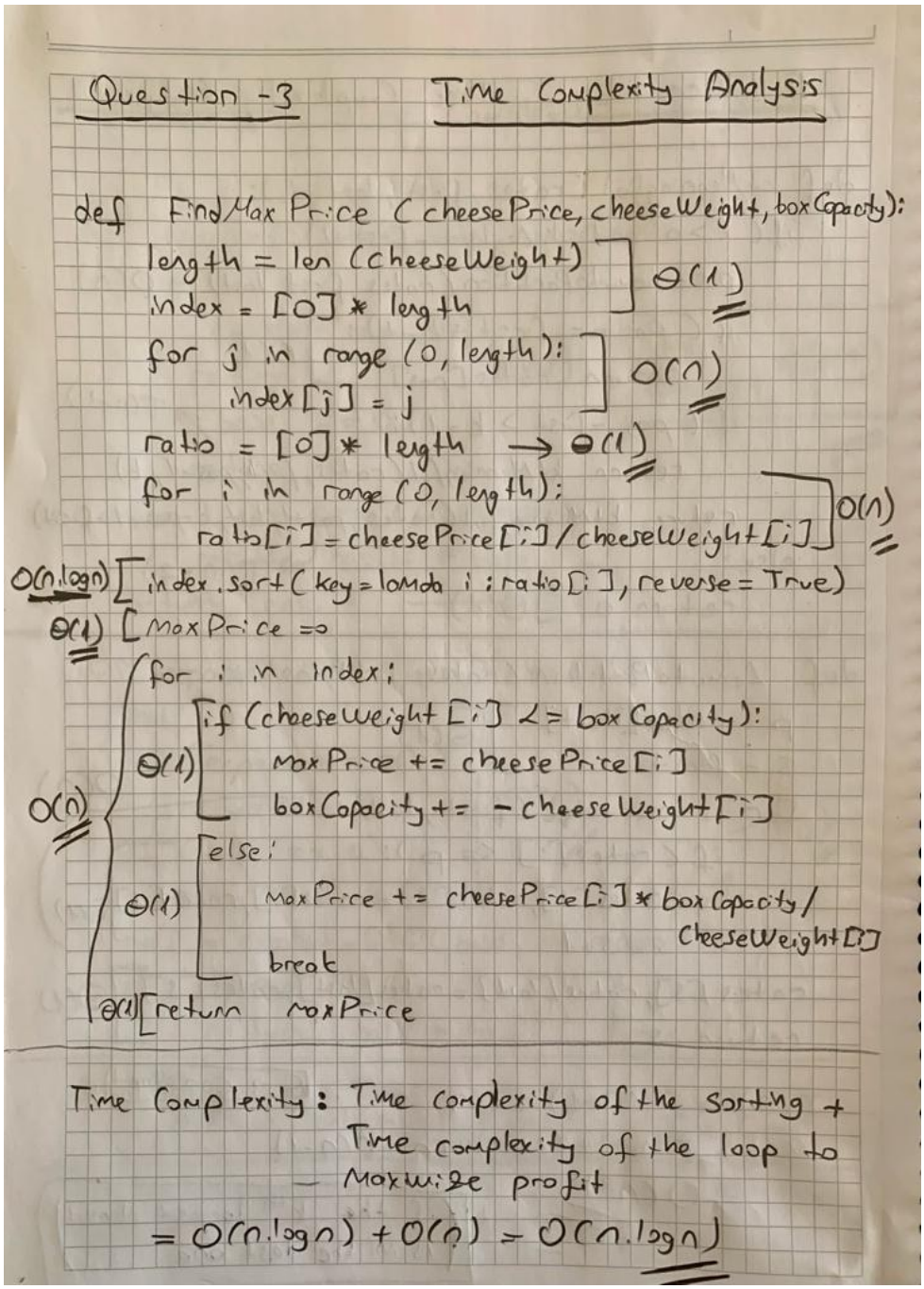
The recursive relation of this algorithm is : maxObtainable = max(arr[j] + temp_arr[i-j-1], maxObtainable).using via buttom to up method,By iterating through the array, The values of the array given to the function as a parameter up to the index value of the outer loop (i) and the values of the temporarily created array from the index value of the outer loop (i) to the 0th index value are added. The result obtained is compared with the variable maxObtainable and assigned to the variable maxObtainable, whichever is greater.

An array is created for temporary use and filled with 0 up to the index value of the array given as a parameter to the function. An outer loop starts from the first index of the array given as a parameter to the function, up to one more than it's the last index.
The variable maxObtainable is set as the maximum negative number for subsequent comparisons in each outer loop. Later, an inner loop is created that will progress by the counter value of an outer loop, and the recursive part of dynamic programming is performed as I explained above. At the end of the inner loop, the variable maxObtainable is assigned to the current index of the temporary array. As a result of all these operations, the last element of the temporary array is returned.



Question – 3

The algorithm takes three arguments: cheesePrice, cheeseWeight and boxCapacity. Algortihm return maxPrice variable where maxPrice is the maximum price of cheeses with total weight not more than box capacity. The function works by choosing an cheese from the remaining cheeses that has the maximum price to weight ratio.If the Box can include the entire weight of the cheese, then the full amount of the cheese is added to the Box. If not, then only a cut of this cheese is added (The Greedy Part) such that the box becomes full.The above three steps are repeated until the Box becomes full, So the total weight reaches the maximum weight.



Question – 4

At the start of the algorithm, The student always chooses the first course because the first course with the earliest start time.After the student takes a course, it is checked whether the student can take another course as follows:

The finish time of the previous course must be less than or equal the start time of the new course to be taken (The Greedy Part) .When this condition is met, the index value that gives the finish time of the course becomes the index value that gives the start time of the new course taken. When this rule is followed, the student can take the course. This is controlled by a loop, If the number of courses taken by the student is equal to 4, since he cannot take any more courses, the loop is exited and no more controls are made.