Q1

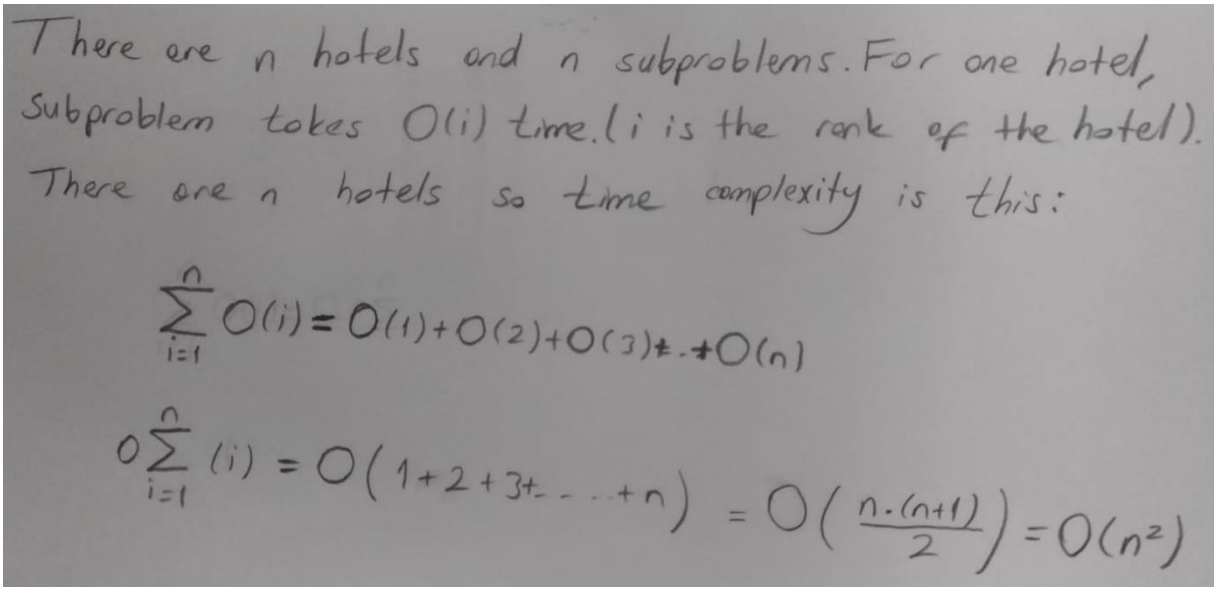
Explaining The Algorithm

I made a dynamic programmic algorithm. I proposed a subproblem which is minimumPenalty(i,a).input i is the hotel rank and input a is the hotels distance array.

Subproblem is finding minimum penalty from starting hotel to hotel i. So I use minimumPenalty function recursively. Also its base case is when i is 0 so it returns 0 and empty path. Becasue there is no hotel.Firstliy , I calculate the previous day stopped hotel and calculate the penalty after sum with today penalty and I choose minimum one.I store the hotel path in an array.

Time Complexity Analyze => O(n^2)

There are n hotels and n subproblems.For one hotel,subproblem takes O(i) time.(i is the rank of the hotel) . There are n hotels so time complexity is O(n^2)

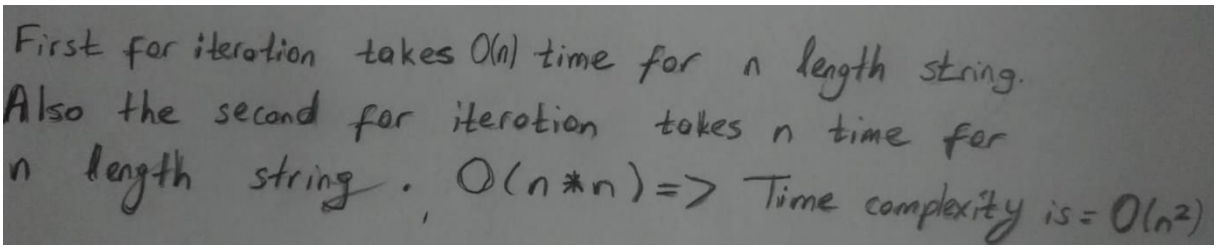


Q2

Explaining The Algorithm

I made a dynamic programmic algorithm. I proposed a subproblem which is finding a result for string s[0....i-1] and then storing it in the resultsOfSubproblems[i]. If result is true for the sub problem s[0....i-1], I store true in resultsOfSubproblems[i]. Otherwise I store false.I traverse the string and search the possible words (which is false in the resultsOfSubproblems list) in the dictionary and if I find it I make it true . When I reach end of the string and if last index of the resultsOfSubproblems ,then I return true otherwise I return false.

Time Complexity Analyze => O(n^2)

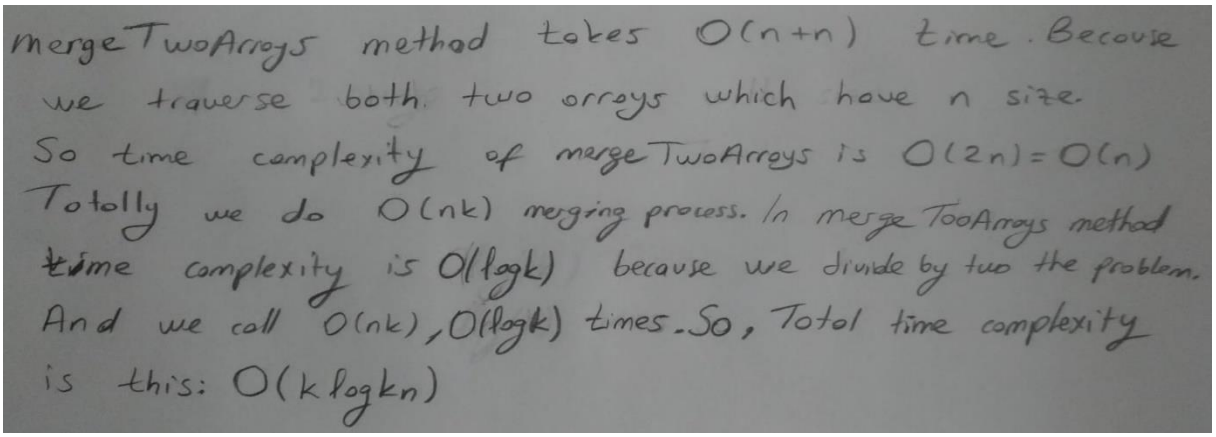


Q3

Explaining The Algorithm

Firstly I take array of array as a parameter in the mergeTooArrays method. If there is one array in the array then I return it(base case of recursive method). Otherwise I divide the array by two and send both two array same method recursively.Now there are two array and I send these arrays to mergeTwoArrays method for merging these two arrays and return this function output as a result of all program. In mergeTwoArrays method I traverse both two arrays and compare each indices for these two arrays. If first array element is smaller than the second array(in the same indices) then I store the first array element in the result array and increment the only first and result array indices. If I reach end one of these two arrays then I continue with the other array and store the element of this array to result array. Lastly I return the result array.

Time Complexity Analyze => O(k*logkn)



Q4

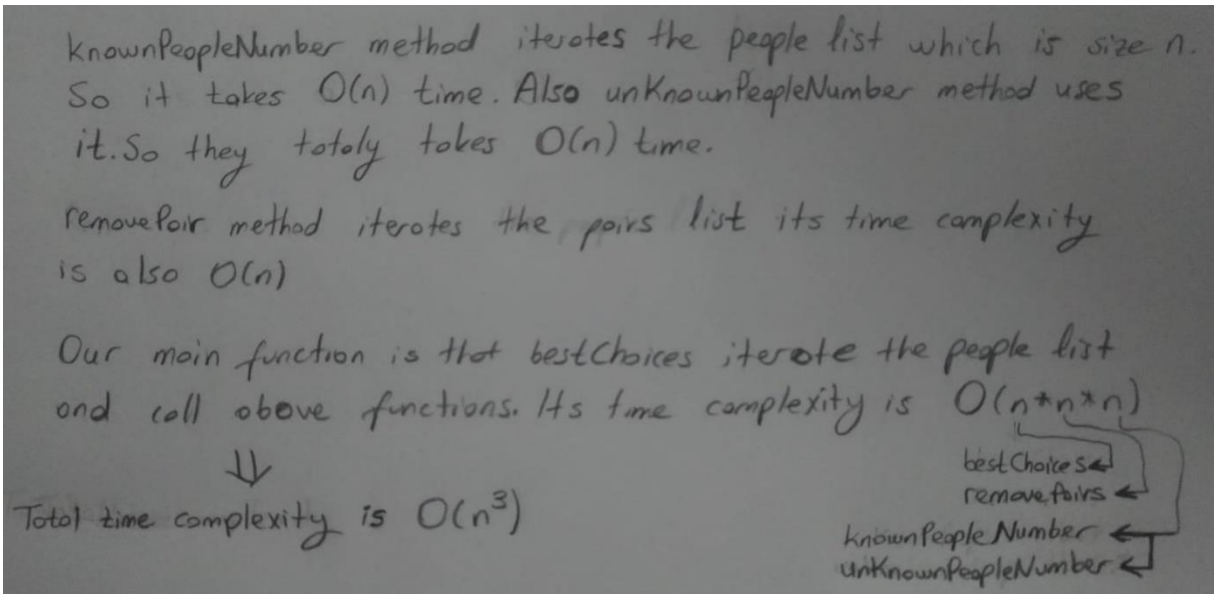
Explaining The Algorithm

I traverse the people list and if I see a person which is know less than 5 people or not know less than 5 people, then I remove this person from the people list and also I remove the pairs which has this person in the pairs list.Because we can’t choose this person for the party.

After that if I remove a person and pairs then I iterate again the people list beacuse this persons can effect the previous persons which have relation with this person.

I all rules is provided so we don't need to remove a person and pair we only increment the people list index and contiune to iterate the list.

Time Complexity Analyze => O(n^3)



Q5

Explaining The Algorithm

getIndexesAndEquality method is an utility method for getting the infos from the constraints string.

Our main method is isSatisfied and in this method I iterate the constraintsList and I get the infos of constraint. With this infos I control this constraint is satisfied or not in the elementList.

If I see a constraint which is not satisfied then I return false otherwise I continue the iterating the constraintsList.

If all the constraints are satisfied then return true.

Time Complexity Analyze => O(m)

We iterate all the constraintList and its size is m. So it takes O(m) time. We don't need to use elemetList for calculating the time complexity. Because we directly access the element with using the indices of the constraints.