INTRIDUCTION TO ALGORITHM DESIGN AND ANALYSIS

HOMEWORK 4 REPORT

PART 1

In this part, I designed and implemented dynamic programming algorithm that solves said problem. Algorithm detemines the optimal sequence of hotels at which stop with minimum penalty.

Complexity

We have n subproblems and each subproblem takes time O(i)

The overal comlexity is

$$\sum_{i=1}^{n} O(i) = O(\sum_{i=1}^{n}) i = O\, n \frac{(n-1)}{2} = O(n^2)$$

PART 2

In this part, I designed and implemented dynamic programming algorithm that solves said problem. Algoritm determines whether given string valid or not and finds reconstructed string using valid words dictionary.

Comlexity

Assume that length of the given string is n. We traverse legth of valid word time for each element in given string.

In worst case we have one valid words, so complexity is n times n.
wost case complexity = $O(n^2)$

PART 3

In this part, I designed and implemented divide and conquer algorithm that solves said problem. Algorithm gives k sorted arrays of array and combine them into single sorted array.

Complexity.

This algorithm is like merge sort algorithm, divides array into two subarray, solves problem for this two subarray and merges them into single array.

Assume that k is number of sorted araays and n is length of each array.

Time comlexity T(k) = T(k/2) + T(n*k/2) = **O(nk.log k)**

PART 4

In this part, I designed and implemented greedy algorithm that solves said problem. Algorithm takes as input known pairs matrix. People are labeled from 0 to n-1(n:number of people). Known matrix is square matrix that contains known and unkonown information. If matrix[i][j] == 1ü ith person knows, jth person, if it is 0, ith person doesn't know jth person. Algorithm first places all people invitee list and cheks known information of each person in invitee list with other person. If number of known people or number of unkown people is less than 5, removes this person from invitee list.

Complexity Analysis

PART 5

In this part, I designed and implemented greedy algorithm that solves said problem. Algorithm takes set of varibles , equality constraints and inequality constraints as input and determine whether given constraints are satisfiable.

Complexity Analysis

The running time is O(n^2). There are O(n) iterations and each we need to scan through invitee list most n remaining possible invitees to find if there is invitee person that violates constrain.

n: number of elements in set
m: total number of equality and inequality constraints

I used union find data structure to solve this problem. Algoritm first assign each varible to its own set and then check whether two elements of an equality $x_i = x_j$ are already in same set using find operation. If they are not, applies union operation. Most n – 1 union operations take O(nlogn) time. Scanning through the list of equalities and applying at most find operation takes O(m) time.
After each equal variable has been placed in the proper sets, we scan through the list of ineuqality constraints. The constraints can be satisfied if, and only if, for every inequality constraints are in different set. So, we check this in linear time. Thus, total running time is **O(m+nlogn).**