CSE 321 Homework 5

141044084

Seyit Ahmet KARACA

1) Algorithm Analysis
   Time complexity of algorithm is O(n) , n is number of jobs
   There are are two loops in method. First loop 0 to number of jobs and it's sorting job list.
   Job list has weights and times so this method's complexity is O(n*2). (2 is fix).
   Other loop is repating n times and it's calculate the weight sum of completion times.
   So method's total complexity is O(n).

Algorithm Explanation

There are a list holds lists which is including weight and times. To minimize the weight sum of completion times , the list must be sorted decreasing order according to w/t. After sorting , the formula is (job[0]'s hour*job[0]'s weight) + ... +((job[i]'s hour + hour from job[0])* job[i]). The formula gives minimize the weight sum of completion times of jobs.

Method:

minimumWeightSumOfCompletionTime(list) : Takes an list as a parameter and returns minimized the weight sum of completion time.

Parameter must be like this ;
jobs = [[5,1000] , [4,10],[4,10]] first one is weight second one is time.

2-a) Suppose that n is 4 and M is 20. Si={1,3,1,8} Ni={20,1,20,5]

The algorithm gives 1+(1+20)+(1+20)+(5+20) = 68. İt's wrong answer because correct answer is 13.

2-b) There is a one loop in method.Loop is repeating number of month.So, time complexity of algorithm is O(n) , n is number of month.

The main formula for this algorithm is :

İ is 0 to number of months, OPTs is array of optimum costs of SF, OPTn is array of optimum costs of NY, M is costs of moving between cities, N is costs of NY, S is costs of SF

OPTn[i] = N[i] + min(OPTn[i-1] , M + OPTs[i-1])
OPTs[i] = S[i] + min(OPTs[i-1] , M + OPTn[i-1])
Result is smallest last element of between OPTn and OPTs