ALGORITHMS MAKE-UP EXAM

Question 1. A traveler wants to walk a road of a given distance *n*. There is a cafe on his/her road placed in **every** discrete location *i* ∈ *{1, 2, …, n}*. The traveler has to rest and have a coffee at some of these cafe locations all of them having have changing coffee prices depending on **the distance he has walked since the last location he rested**.

The prices for all possible distances are stored in a *price* array. That is, *price[1]* denotes the price that the traveler has to pay for the coffee if s/he walked a distance of *1* since the last location s/he rested*, price[2]* denotes the price that he has to pay if s/he walked a distance of *2* since the last location s/he rested and so on. For example; when *n = 9*, if s/he firstly rests at location *3*, s/he has to pay *price[3]* and then, if s/he rests at location *5*, s/he has to pay *price[2]* and then, if s/he rests at location *9*, s/he has to pay *price[4]*. Hence, the total price makes *"price[3] + price[2] + price[4]"*.

Your goal is to minimize the total price s/he has to pay.

For example; when *n = 4* and *price = [3,2,5,9]*, the optimum locations that s/he has to rest are *2* and *4* as shown in the table below.

Stop locations	**Total price**
4	*9*
3,4	*5+3=8*
2,4	*2+2=4*
1,4	*3+5=8*
2,3,4	*2+3+3=8*
1,3,4	*3+2+3=8*
1,2,4	*3+3+2=8*
1,2,3,4	*3+3+3+3=12*

Given a distance *n* and a *price* array,

a. (20pts.) Give a mathematical recursive formulation for *P(n)* where *P(n)* denotes the minimum price the traveler has to pay when he wants to walk a distance of *n*.

b. (20pts.) Write a recursive algorithm (i.e., a pseudocode) that returns the minimum price the traveler has to pay when he wants to walk a distance of *n*.

c. (20pts.) Give an example distance *n* and a *price* array to show that the greedy choice of every time choosing to walk the distance, which does not exceed the total distance *n* and would cost the minimum price at the next stop location, does not always lead to an optimum solution.

d. (30pts.) Write a dynamic programming algorithm (i.e., a pseudocode) for finding the minimum price the traveler has to pay to walk a distance of *n*.

e. (10 pts.) Provide the running time of your dynamic programming algorithm. Explain.

P.S. It is extremely necessary to understand the question (e.g. what the *price* array holds).