

Introduction To Algorithm Design

Homework 5

Report File

Mustafa Tokgöz

171044077

Explanation of Question 1)

S is list in the python code ; i, length variables in python code.

I design an algorithm that find the sum of subsets that equals to 0. The algorithm firstly checking if the current position is equal to size or not. if it is then it is adding i th element of subset to sum i to length with loop. Then if sum is equal to zero then it shows the elements of subset. If current position is not equal to size then it assigns array[current position] to S[length] for subset. Then it is calling algorithm_question5 function recursively by adding 1 to current position and length. Why i use the second algorithm_question1 call is to skip the current element by adding 1 to only current position.

Explanation of Question 2)

Dp = list in python code ; i , j variables in python code

The algorithm starts from bottom to top. The algorithm first assigns bottom row to dp list with loop. After that from bottom to up, if j th element of dp list is less than the one forward element then algorithm stores i th row and j th element of list arr by adding j th element of dp list to j th index dp list in loop column number times. This part of code is in loop row number times. For example let triangle is [2], [5 ,4] then $dp[0] = 5$ and $dp[1] = 4$ after that $dp[0] > dp[1]$ and $dp[0] = 2 + 4$ after that returns $dp[0]$. Also it adds subset elements to list

then print it. So it finds smallest sum path. I print smallest path and sum of it.

Explanation of Question 3)

arr = list in the python code ; i,j,wight,value variable in python code.

The algorithm firstly checking j th weight is less or equal than i or not in the inner loop. If less, then also if i th element of arr is less than arr [i - j th element of weight addition with value of j th element] , Then it stores the value of arr [i - j th element of weight addition with value of j th element] to i th index of arr. After that it returns the W th element of array arr. This problem is different from the original knapsack problem because in this problem we can can pick from each item as many as we can.