ASSIGNMENT D3

Combinatorial Algorithms for CS4B

Submitted By: Ahmed Dider Rahat

Matriculation Number: 916146

Assignment Code: After implementing all the assignment I pushed my code on my Github link.

Answer to the question no. 1

The highest possible benefit is 375.

Answer to the question no. 2

The goal: Get highest benefit.

Constraint: Can use maximum cost of 200 unit

The given benefit, cost pairs are: `(50, 10),(25, 20),(30, 10),(15, 80),(30, 60),(30, 20),(30, 20),(120, 30),(30, 10),(30, 25),(30, 25).`

My computation steps are:

1. Remove two big cost data of 60, and 80. So, the list become: `(50, 10), (25, 20), (30, 10), (30, 20), (30, 20), (120, 30), (30, 10), (30, 25), (30, 25).`
2. Sort with respect to benefit: `(120, 30), (50, 10), (30, 10), (30, 20), (30, 20), (30, 10), (30, 25), (30, 25), (25, 20).`
3. Calculate the cost.

   `Cost = 30+10+10+20+20+10+25+25+20 = 170` → `Satisfied the cost condition`
4. Calculate the benefit:

   `Benefit = 120+50+30+30+30+30+30+30+25 = 375` → `Final Output`

Answer to the question no. 3

Limiting factors of my approach:

1. I elemenate the two large element just intutionally. So, the approach is not always produce good result.
2. Sort the element just using their cost aslo may not always prodecure optimal solution.
3. If the data set become larger, its really tough to compute the optimal solution.

Answer to the question no. 4

The underlying mathematical or combinatorial conceptis power set. Because we have to generate all possible solution of the original tasks.

If we consider the set of tasks are: $S = \{0, 1, 2, 3,4,5,6,7,8,9,10\}$

So, the all possible solution inclued:

$$allpossible(S) = \{\, Ø, \{0\}, \{1\}, \ldots, \{10\}, \{0, 1\}, \{0,2\} \ldots, \{0,10\}, \ldots \{0,1,2, \ldots ,10\} \,\}$$

Here, all possible(S) is the power set of S.

Answer to the question no. 5

The implementation of this question are given in ‘Question_5.py’ file.

Answer to the question no. 6

If the given input is 0 to 2 then the function return 2^3 output sequence. The function has a loop inside it. The loop execute for each of the data point. And, each times it execute a list comprehension inside it which runs for . So, the simulation become:

When execute 0, it return an array with [0]
When execute 1, it return an array with [1], [0,1]
When execute 2, it return an array with [2], [0, 2], [1, 2], [0, 1, 2]

Answer to the question no. 7

For the given task, with 11 elements the program create 2^11 = 2048 list items. And for n element the items should be 2^n.

Answer to the question no. 8

`‘Yield’` returned the generator of the list that have been prodecure on the otherhand `’return’` statement returned the whole object.