27096661 Yanda Qu

Task1:

Idea: using DP, we need to store the past results. Therefore, using a list called "memo" to store the max profit for price limit from 0 to the limit you enter. Therefore, as we know the max profit of price limit n must come from one item plus its corresponding past answer. For price limit n, go through every item and check if one item can be chosen to add from lower price limit. For every item chosen, let "index" be the current price limit minus the price of the item, get the profit at position "index" and plus the profit of that item. Then compare all the possible profit of one price limit, choose the maximum and store the item chosen in a decision list. After looping, the last element of the memo list will be the max profit. Then using the decision list, going through it from the last element, which is the last item we choose. Every time the loop computes, let the index of the position we currently focus in the decision list minus the price of the item which is currently choose, this will give us the index of the item we previously choose in the decision list. In the end, the content in the index we focus will reach -1 because the limit will finally be smaller than the price of any item in the products list. Now, the items we choose are all stored in a solution list. Go through solution list to print result.

Complexity: the read file function read file of size N into python which take O(N) time and space. Then, we go through all the price limit from 0 to P and inside this loop we go through every item N therefore this has O(NP) time. All the space we need for this is a memo list and a decision list which means it has space O(N+P). to print the result. We go through decision list so time is O(P), we create a new solution list therefore, the space is O(P) as well. Overall, the time complexity is O(NP) and the space complexity is O(P+N).

Task2:

Idea: like task1 however, this time we use a table to store past result. Table memo has rows labelled "item limit" and columns labelled "price limit". The idea is for every item limit, we calculate the max profit for all the price limit according to the previous row which is the row has lower item limit. This is because for an item limit with one more item, the profit of one price limit must come from the profit of smaller price limit of one less item limit. Every time calculate a new price limit, go through all the items and check if one item can be added. If can, we add the profit of the item to the profit store in the index which is current index minus the price of the item in the previous row. Using this, we can calculate all the vertex in the table. Then the right corner of the table is the maximum profit we want. Because every time the item we choose is store in the decision table. Then we can use similar method in task1 to print the result.

Complexity: like task1, however this time we have item limit. This let we go through every item limit therefore to get maximum profit need O(MNP) time where M is the item limit. We use space saving method so memo is a two-row table of length P but we use a decision table instead, then the space is O(MP). To print the result, we go through all the output where the output is at most of size of the item limit which is O(M) and the solution list is created with O(M) space as well. Overall the time complexity is O(MNP) and the space complexity is O(MP).