NAME: Piyush Mahatkar.

NET ID: PKM170230.

CS 6363.004 Algorithms

Programming Project:

Solution:-

* Recurrence:-

$$
\text{Profits}_{(i,g)} = \begin{cases} 0 & \text{if } i=0 \\ \text{Profits}(i-1,0) - \min(f_i * n_i, c_i) & \text{if } g=0 \\ \max\limits_{0<q\le x_i} \begin{cases} \text{if } n_i > q \\ \quad q * p_i + \text{profits}(i-1, g-q*w_i) \\ \quad\quad - \min(f_i * (n_i - q), c_i) \\ \text{else} \\ \quad q * p_i + \text{profits}(i-1, g-(q*w_i)) \end{cases} \end{cases}
$$

* Conversion to Dynamic Program:

```
1  for i = 1 to n, do
2          for j = 1 to G do
3                  profits[i][j] = -∞
4                  for q = 1 to items[i].max
5                          Profit = 0
```

Scanned by CamScanner

```
if q < items[i].min
        Profit = q.Pi + Profits[i-1][j - q * wi]
                 - min [ fi * (ni - qi), ci]
    else
        profit = q . Pi + profits[i-1][j-q * wi]

    Profits[i][j] = Profit.
  end for_j
end for_i
end.
```

* Proof of Correctness:

Feasibility: There is a solution profits(n, G) when G units of gold is used to make items that yeilds maximum profit.

→ case: Base case:

$G=0$ and $i=0$

Profits$[i][j] = 0$, true and valid.

Proof by induction :-

Let us assume it is valid for k items, with $G + n > 0$

Case 1:

~~ifcase~~, $n_i > q$.

$$profits(K, g) = \max\left[q \cdot P_k + Profits(k-1, g - q \cdot w_k)\right.$$
$$\left. - \min\left(f_k \cdot (n_k - q), c_i\right)\right.$$

there is a solution.

Case 2: $q > n_i$

$$Profits(i, k) = Max\left[q \cdot P_k + Profits(k-1, g-(q \cdot w_k)\right.$$

Solution ~~is there,~~ exists.

Case 3:

if $G = 0$

$$profits(K, g) = profits(K-1, 0) - \min(f_i \cdot n_i, c_i)$$

$\therefore$ There is a solution which gives a solution.

$\therefore$ It is feasible.

* Optimality:

let $opt(q, G)$ be an optimal solution

$G \rightarrow$ gold units $q \rightarrow$ items.

we say: $opt(i, G) \leq Profits(i, G)$.

Scanned by CamScanner

Proof by induction:-

base case: G & $i = 0$ i.e $G + i = 0$

$opt(0,0) = Profits(0,0)$.

Step 1: $i + G > 0$,

Case 1: $q < n_i$

$$Opt(i, G) = \overset{q \cdot P_i}{\hat{}}\, Opt(i-1, G - q \cdot w_i) - \min(f_i(n_i - q), c_i)$$

$$Profits(i, G) \geq \overset{q \cdot P_i}{\hat{}}\, Profits(i-1, G - q \cdot w_i) - \min(f_i(n_i - q), c_i)$$

$\therefore$ by induction, $Profits(i, G) \geq opt(i, G)$.

Case 2: $q \geq n_i$

$$Opt(i, G) = opt(i-1, G - q \cdot w_i) + \min(q \cdot P_i)$$

$$Profits(i, G) \geq Profit(i-1, G - q\, w_i) + q \cdot P_i$$

as, $(G - q\, w_i, i-1) < i + G$

$\therefore$ we can say by IH

$Profits(i, G) \geq opt(i, G)$

$\therefore$ Proved by IH.

Scanned by CamScanner

* Runtime Complexity and Analysis:

loop at line (1) → n times
loop at line (2) → G times
loop at line (4) → items[i^{th}].max.

$$RT = O(n \cdot G + items.max).$$

Scanned by CamScanner