

Programming Project

* Recurrence for Maximum Profit

$$\gamma(i, k) = \begin{cases} 0 & \text{if } i=0 \\ \max_{0 \leq q_i \leq x_i} \left(p_i q_i - \min(f_i(n_i - q_i), c) + \gamma(i-1, k - w_i q_i) \right) & \begin{array}{l} \text{if } i > 0, \\ \text{if } n_i > q_i \\ \text{if } k - w_i q_i \geq 0 \end{array} \\ \max_{0 \leq q_i \leq x_i} \left(p_i q_i + \gamma(i-1, k - w_i q_i) \right) & \begin{array}{l} \text{if } p_i > 0, \\ \text{if } n_i \leq q_i \\ \text{if } k - w_i q_i \geq 0 \end{array} \end{cases}$$

$i = i^{\text{th}}$ item.

w_i = weight of gold required for the i^{th} item.

p_i = profit of one quantity of that i^{th} item.

n_i = minimum quantity of i^{th} item required / lower limit

x_i = maximum quantity of i^{th} item / upper limit

f_i = fine for one quantity deficit

c_i = cap for the fine of that item.

k = local gold available.

* Proof of correctness

① Feasibility \rightarrow There is a solution $\gamma(i, k)$ that has some profit for i is in range of items & k is in range Gold

By Induction on $i+k$.

② Optimality \rightarrow let $\text{opt}(i, k)$ be the optimal solution to the problem

Claim $\rightarrow \gamma(i, k) \geq \text{opt}(i, k)$.

we do this by doing Induction on $i+k$.

• Base Case $i = 0$.

this will be the zero item or just a dummy item.
everything corresponding to $i=0$ & k with any value
will be zero.

In this case the optimal solution will be zero.

$$\text{∴ } \text{opt}(0, k) = 0 \quad \text{for any value of } k \\ 0 \leq k \leq G_1.$$

And,

$$\gamma(0, k) = 0 \quad \text{for any value of } k \\ 0 \leq k \leq G_1.$$

$$\text{∴ for } i=0 \text{ & } 0 \leq k \leq G_1 \\ \gamma(0, k) = \text{opt}(0, k) = 0.$$

Base case true.

• Step $\rightarrow i+k > 0$. Or $i+k \neq 0$.

here we have to show if, for any value of i & k

$$\gamma(i, k) \geq \text{opt}(i, k).$$

lets say that the opt take q_i^* copies of an item i^{th}

then

$$\text{opt}(i, k) = p_i q_i^* - \min(f_i(n_i - q_i^*), c_i) + \text{opt}(i-1, k - w_i q_i^*)$$

&

$$\gamma(i, k) = \max_{0 \leq q_i \leq x_i} \left(p_i q_i - \min(f_i(n_i - q_i), c) + \gamma(i-1, k - w_i q_i) \right) \text{ if } k - w_i q_i > 0$$

lets make the same cut in our recurrence.

$$\gamma(i, k) \geq p_i q_i^* - \min(f_i(n_i - q_i^*), c_i) + \gamma(i-1, k - w_i q_i^*)$$

because now we are using the optimal cut

(3)

now in $i-1 + k - w_i q_i^*$

there will be two cases where.

Case 1 $q_i^* = 0$.

$$i-1 + k - 0 < i+k$$

$$\therefore \cancel{x(i-1, k-w_i q_i^*)} \geq \text{opt}(i-1, k-w_i q_i^*) \quad \text{--- (2)}$$

--- done.

Case 2 $q_i^* \neq 0$. or $q_i^* > 0$

here as well.

$$i-1 + k - w_i q_i^* < i+k$$

$$\therefore x(i-1, k-w_i q_i^*) \geq \text{opt}(i-1, k-w_i q_i^*)$$

--- done.

\therefore now we can say that

$$x(i, k) \geq p_i q_i^* - \min(f_i(n_i - q_i^*), c) \\ + \text{opt}(i-1, k-w_i q_i^*)$$

--- By Induction Hypothesis.

from (1) (2) (3) (4).

$$x(i, k) \geq \text{opt}(i, k).$$

we can do the same with the recurrence base.

where there is no fine. i.e. if $n_i < q_i^*$

we just have to remove the $\min(f_i(n_i - q_i^*), c)$ from the above proof.

* Dynamic Programming Algorithm for calculating Maximum profit

→ profit

for $i = 1$ to n

for $k = 0$ to G_i .

localMax = 0.

for $q_i = 0$ to x_i

if $k - w_i q_i \geq 0$

if $n_i > q_i$:

// deficit

$$x(i, k) = p_i q_i \cdot \min(f_i(n_i - q_i), c_i) + x(i-1, k - w_i q_i)$$

else:

// no deficit

$$x(i, k) = p_i q_i + x(i-1, k - w_i q_i)$$

n → number of items

n_i → Item at i^{th} index. ~~minimum~~ quantity.

G → Total Gold.

q_i → quantity of item.

x_i → ~~maximum~~ quantity required of i^{th} item.

p_i → profit from one quantity of item i .

f_i → ~~free~~ free for i^{th} item.

w_i → weight of i^{th} item.

$x(n, G)$ will give the Maximum profit.

* Recurrence for count of different optimal Solutions.

$$\text{cnt}(i, k) = \begin{cases} 1 & \text{if } i=0 \\ \text{Sum}(\text{cnt}(i-1, k - w_i q_i)) & \text{if } \text{opt}(i, k) = p_i q_i \\ 0 \leq q_i \leq x_i & - \min(f_i(n-q_i), c_i) \\ k - w_i q_i > 0 & + \text{opt}(i-1, k - w_i q_i) \end{cases}$$

or

$$\text{opt}(i, k) = p_i q_i + \text{opt}(i-1, k - w_i q_i)$$

i = i^{th} item.

w_i = weight of gold required for i^{th} item.

p_i = Profit of one quantity of that i^{th} item.

n_i = minimum quantity of i^{th} item required / lower limit

x_i = maximum quantity of i^{th} item ~~is~~ / upper limit

f_i = fine for one quantity deficit

c_i = cap for the fine on that i^{th} item.

K = local gold available.

* Proof of Correctness.

① Feasibility \rightarrow There is a solution $\text{Cnt}(i, k)$ that gives the count of no. of optimal solutions.

By Induction on $i+k$.

② Optimality \rightarrow let $\text{Opt}(i, k)$ be the optimal solution to the problem.

Claim $\rightarrow \text{Cnt}(i, k) \geq \text{Opt}(i, k)$

We will do this by doing Induction on $i+k$.

- Base Case. $\text{Opt}^*(0, k) = 0$

even there are no items or just a dummy item we have.
 $\text{Opt}(0, k) = 0$, which is the only answer.

$$\text{so } \text{Opt}(0, k) = \text{Cnt}(0, k) = 1$$

\therefore Base Case true.

- Step. $i+k > 0$

here we have to show if for any value of i, k .

$$\text{Cnt}(i, k) \geq \text{Opt}(i, k)$$

lets say opt takes q_i^* copies of an item.

then

$$\text{Opt}(i, k) = \text{Sum}(\text{Opt}(i-1, k - w_i q_i^*)) \quad \text{--- (1)}$$

q_i^* being all
optimal solution,
with q_i^* copies
of i^{th} item.

lets try the same in our recurrence

$$\text{Cnt}(i, k) \geq \text{Sum}(\text{Cnt}(i-1, k - w_i q_i^*)) \quad \text{--- (2)}$$

because we are using the same as
opt

now in $i-1 + k - w_i q_i^*$ there will be two cases,

unless .

Case 1 $q_i^* = 0$

$$i-1 + k - 0 \leq i+k$$

$$\therefore \text{Cnt}(i-1, k - w_i q_i^*) \geq \text{Opt}(i-1, k - w_i q_i^*) \quad \text{--- (3)}$$

done,

Case 2

$$q_i^* > 0.$$

here again

$$v_{i-1} + k - w_i q_i^* < v_i + k.$$

$$\therefore \text{Cnt}(i-1, k - w_i q_i^*) \geq \text{opt}(i-1, k - w_i q_i^*)$$

done

(7)

from ① ② ③ ④

$$\text{Cnt}(i-1, k) \geq \text{opt}(i, k)$$

We do same for deficit & no deficit case.

returning $\text{cnt}(n, G)$ will give number of optimal solutions.

* Dynamic Programming Algorithm for counting number of optimal solutions.

for $i = 1$ to n .

for $k = 0$ to G_i

for $q = 0$ to x_i

if $k - w_i q_i \geq 0$

if $n_i > q_i$:

deficit

$$\text{cnt}(i, k) = \text{cnt}(i, k) + \text{cnt}(i-1, k - w_i q_i)$$

else:

no deficit

$$\text{cnt}(i, k) = \text{cnt}(i, k) + \text{cnt}(i-1, k - w_i q_i)$$

n = no. of items.

n_i = minimum quantity of item i required

G_i = Total Gold.

q_i = Quantity of item i .

x_i = maximum quantity of i th item.

p_i = Profit from one quantity of item i .

f_i = fine for i th item.

w_i = weight of i th item.

$\text{cnt}(n, G)$ will give number of optimal solutions

RT Analysis → .

(8)

Since we are running three loops.

- ① $i \rightarrow 0$ to n .
- ② $k \rightarrow 0$ to G_1
- ③ $q \rightarrow 0$ to x_i

So the Running Time will be multiplication of.

$$n \times G_1 \times x_i$$

But different items will have different amount upper limit, so in this case. the $\max(x_i)$ will dominate

$$\therefore RT = O(n \times G_1 \times \max(x_i))$$