01 Knapsack problem

2017년 10월 18일 수요일 오후 3:24

Quiz : LIS

A 1 10 2 5 3 7 8 9 4

L(i) 1 2 2 3 3 4 5 6 4

B 2 11 3 6 4 8 9 10 5

$L(n) = 4$

$\max\{L(i)\}$

$L(i) = 1 + \max\{L(j) : s_j < s_i\}$

$O(n^2)$

Given a set of items, I, indexed from 1 to n, where item I has weight w_i>0 and profit p_i and a knapsack capacity of M,
find the subset I' of I that maximized the total profit such that the sum of weights of selected items in I' are smaller or equal to M.

Let R_l^i be maximum profit possible using a subset of elements indexed {1,…, *i*} and yield weight exactly *l*.
(i<=n, 0<=l<=M)

R_M^n

$R_l^0 = 0$ for $0 < l \le M$

$1 < i \le n$

Recurrence step:

$$R_l^i = \max\{R_l^{i-1}, R_{l-w_i}^{i-1} + p_i\} \text{ for } 0 \le l \le M$$

Maxtrix-based Algo

O(M) for $l \leftarrow 0$ to M do $R_l^0 = 0$ endfor

O(n) for $i \leftarrow 1$ to n do $R_0^i = 0$ endfor

O(nM) for $i \leftarrow 1$ to n do

for $l \leftarrow 0$ to M do

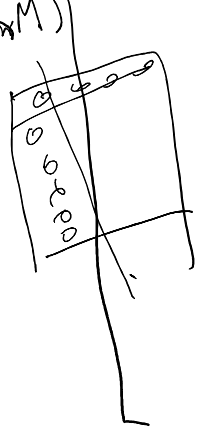
if $w_i \le l$ and $R_{l-w_i}^{i-1} + p_i > R_l^{i-1}$

then $R_l^i \leftarrow R_{l-w_i}^{i-1} + p_i$ /* take i */

$B_l^i \leftarrow 1$

for $i \leftarrow 1$ to n do

$O(nM)$



for $l \leftarrow 0$ to M do

if $w_i \le l$ and $R^{i-1}_{l-w_i} + p_i > R^{i-1}_{l}$

then $R^{i}_{l} \leftarrow R^{i-1}_{l-w_i} + p_i$ /* take i */

$B^{i}_{l} \leftarrow 1$

else $R^{i}_{l} \leftarrow R^{i-1}_{l}$

$B^{i}_{l} \leftarrow 0$ /* not take i */

endif

endfor

endfor

$O(n) + O(M) + O(nM) = O(nM)$

$k = M$

for $i = n$ downto 1 do

if $B^{i}_{k} = 1$ then

print i

$k \leftarrow k - w_i$

endif

~~for~~ endfor

return R^{n}_{M}

time $O(nM)$

space $O(nM)$

* list-based Algo

adjust to Basis & Recursion

Basis : $R_0^0 = 0$

Recurrence : $R_\ell^i = \max\{ R_\ell^{i-1}, R_{\ell-w_i}^{i-1} + p_i \}$

undefined case

$R_\ell^i = R_\ell^{i-1}$ if R_ℓ^{i-1} defined but not $R_{\ell-w_i}^{i-1}$

$R_\ell^i = R_{\ell-w_i}^{i-1}$ if $R_{\ell-w_i}^{i-1}$ " but not R_ℓ^{i-1}

$R_\ell^i =$ undefined if Both not defined.

list-based Algo

/* triple (p, w, B) */

p = profit, w = weight, B = Backpointer

/* T^i is set of small {triples} */

$T^0 \leftarrow (0,0,0)$

/* T^i is sorted by p-value hashed by w value */

for $i \leftarrow 1$ to n do

- Generate T_I for T^{i-1} by

not take i → ① copying each triple (p, w, B) from T^{i-1}

not take i → ① copying each triple (p, w, β) from T^{i}

take i → ② and inserting $(p+p_i, w+w_i, i)$

∘ ~~Eliminate T^i from T^{i-1}~~

∘ Scan T_I and delete (p', w', β') from I_I whenever we have (p', w', β') and (p'', w'', β'') where $p' \leq p''$ and $w'' = w'$.

∘ Also remove if $w > M$.

new set : T_i

end for

/* follow the backpointer, from triple in T^n with largest profit */

β = index of the triple with largest profit

While $\beta \neq 0$ do

~~print β~~

Scan T^n to find $\langle P - p_\beta, W - w_\beta, \beta' \rangle$

$n \leftarrow n-1$

end while

2 1