Dp 6 Jan

Knapsack problems

Knapsack problem

Basic idea:
Given n objects, each object has a value & weight associated with it.

value → profit/loss.

Given a bag with capacity W.
Select some items such that:
1) the sum of weights of selected items <= W
2) the sum of values of selected items is maximized/minimized

Variations

1) Traditional Knapsack
→ can take fractions of an item.

0/1 Pack

N objs
knapsack → {2, 3, 4, 5}
weight → {4, 2, 3, 1}

Cap (W) = 5
max = 11

Pick from this find which are bag goods knapsack/sack.

What is the max knapsack you can get?

Greedy fail

Cost | Sellers

W = { 4, 2, 3, 1 }
5 or 5 : 1.25 | 1.5 | 1.33 | 1.0

1.5 | 2.5 | 2.25 | 2.5

Greedy ✓

2) 0/1 Knapsack

Either pick the whole thing or don't

X fractions!

Top down

H → { 4, 2, 3, 1 }
W → 5

Not ↓ | ↓ | ↓ | ↓
Not → | → | → | →

W = 5

Greedy X

N

H | W

X | Bottom

$2^{n-1}$ | W

H | W

$2^{n-1}$ | W-W[n-1]

H-W[n-1] | W-W[n-1]

$\Delta = 2^{n-1} - 1$

X $O(2^n)$

$\rightarrow O(N^2)$

$f_p(i,j)$ = max profit you can get considering items 0 to (i-1) & cap of the bag is j

$\rightarrow O(N \times W)$

$TS \rightarrow n \times W$

$f_p(i,j)$ = max( $f_p(i-1,j)$, $f_p(i-1, j-w[i]) + v[i]$ )

$f_p(i,j)$ = max profit you can get considering items 0 to (i-1) & cap of the bag is j

$f_p[i] = f_p(i, j-w[i])$

$TC: O(N \times W)$

$SC: O(N \times W)$

$\rightarrow O(W)$

Base Case
i == 0 → wt > 0
j == 0 → wt > 0

int state(i, j) {
if(i == 0 && wt > 0)
return 0;
if(j == 0 && wt > 0)
return 0;
if(dp[i][j] != -1) return dp[i][j];
int ans = 0;
if(j >= wt[i])
ans = max(state(i-1, j), p[i] + state(i-1, j-wt[i]));
else
ans = state(i-1, j);
dp[i][j] = ans;
return ans;
}

Find the indexes of the objects that you have to pick from the given problem

if( $f_p(i,j) == f_p(i-1,j)$ )
do not pick the i th item

else
↳ do pick the i th item!

N = 5
W = 5
wt = 5

$f_p(i,j)$ = the max profit we can get considering items 0 to (i-1) and a knapsack of cap = j

int state(int i, int j) {
if(i == 0) {
if(dp[i][j] != -1) return dp[i][j];
int ans = 0;
if(j >= wt[i])
ans = max(ans, p[i] + state(i-1, j-wt[i]));
else
ans = state(i-1, j);
dp[i][j] = ans;
return ans;
}
else if(j >= wt[i])
return p[i] + state(i-1, j-wt[i]);
else
return state(i-1, j);
}

$TC: O(N \times W)$

$SC: O(N \times W)$

$\rightarrow O(W)$

Unbounded Knapsack / 0-N Knapsack

→ ∞ copies of each item!

N => { 0, 1, 2, 3 }
P => { 5, 2, 3, 7 }
W => { 2, 3, 4, 1 }

$f_p(i,j)$ = the max profit we can get considering items 0 to (i-1) and a knapsack of cap = j

$f_p(i,j)$ = max( $f_p(i-1,j)$, $p[i] + f_p(i, j-w[i])$ )

$TC: O(N \times W)$

$SC: O(N \times W)$

0/1 Knapsack with diff capacity

$TC: O(n \times W)$

$1 \le N \le 100$
$1 \le W[i] \le 10^6$
$1 \le P[i] \le 100$
$1 \le W \le 10^6$

$100 \times 10^6$
$5 \times 10^8$

$f_p(i,j)$ → cap
$f_p(i,j)$ → profit

$f_p(i,j)$ → no cap for bag

$f_p(0, 25)$

$f_p(i,j)$ → max

$f_p[n][W]$

$\rightarrow w[i] + f_p(i-1, j-p[i])$

$f_p[n][W]$

$N \le 10$
$100 \le P_i \le 10$
$100 \le W_i \le 10^6$
$1 \le W \le 10^6$

$f_p[n-1][j] \rightarrow$ max cap $\le j$
length of profit considering all items.

$f_p[n-1][25]$ | $f_p[n-1][25-w_n]$ | $f_p[n-1][25]$

$f_p[n-1][j] <= 25$

$N: 3n$