Dinamik Programlama ve Greedy Yaklaşımı

w → ağırlık sınırı kapasitesi

k → adet eleman

$$\underbrace{B[k,w]}$$

Bu durumda ki değerini ifade eder.

Brut Force → n tane eşya var, w kapasitesi aşmadan çantayı en değerli yapmak için alabilecek tüm nesneleri deneriz. n adet nesne için $2^n \rightarrow$ adet alt küme vardır. Hepsine bakarız. Algoritma karmaşıklığı 2^n olacaktır.

Eğer, w_k (seçilen elemanın ağırlığı $< W$ ise ve k. eleman koyulduğunda çanta değeri artıyorsa k. eleman çantaya koyulur.

$$B[k,w] = \begin{cases} B[k-1,w] & \text{if } w_k > W \\ \max\{B[k-1,w]\}, B\{[k-1,w-w_k]+b_k\} & \text{else} \end{cases}$$

~~Ör~~

n → eleman sayısı 4

w → çanta kapasitesi 5

Elemanlar I(w, v) $\underset{\text{ağırlık}}{\overset{1.}{(2,3)}}$ $\underset{\text{değer}}{\overset{2.}{(3,4)}}$ $\overset{3.}{(4,5)}$ $\overset{4.}{(5,6)}$

Taşıma kapasitesi?

→ taşıma kapasitesi?

1. 2. 3. 4 nesne ↓

i \ w	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	i=1 $w_i=2$ $v_i=3$ w=1 ↳ ağırlık aştı $w-w_i=-1$ +4 alamayız 0	+3 i=1 $w_i=2$ $v_i=3$ w=2 $w-w_i=0$ 3 +4	+3 i=1 w=3 $v_i=3$ $w_i=2$ $w-w_i=1$ 0,1 3	i=1 w=4 $v_i=3$ $w_i=2$ $w-w_i=2$ 0,2 i−1 w−wᵢ 3	i=1 w=5 $v_i=3$ $w_i=2$ $w-w_i=3$ 0,3 i−1 w−wᵢ 3
2	0 +5	i=2 w=1 $v_i=4$ $w_i=3$ 1−3=−2 ekleme yok 0	i=2 w=2 $v_i=4$ $w_i=3$ 2−3=−1 ekleme yok 3	i=2 w=3 $v_i=4$ $w_i=3$ 3−3=0 i−1, w−wᵢ 1 0 4	i=2 w=4 $v_i=4$ $w_i=3$ 4−3=1 i−1, w−wᵢ 1,1 4	i=2 w=5 $v_i=4$ $w_i=3$ 5−3=2 1,2 7
3	0 +6	i=3 w=1 $v_i=5$ $w_i=4$ 0	3	4	i=3 w=4 $v_i=5$ $w_i=4$ $w-w_i=4-4=0$ i−1, 0 5	i=3 w=5 $v_i=5$ $w_i=4$ 5−4=1 i=1 2,1 5 7
4	0	0	3	4	ağırlıktan dolayı 5	i=4 w=5 $v_i=6$ $w_i=5$ i−1, w−wᵢ 3, 0 6 → 7

Bu algoritmanın çalışma zamanı $(n \cdot W)$ ← değeri yüksel olabilir

Brute force olsaydı $O(2^n)$

Greedy nlogn ile çözer ama en optimum çözümü vermez