

Dinamik Programlama

ve Greedy Yaklaşımı

$$B[k, w]$$

w - sırt çantası kapasitesi

k - adet eleman

Bu durumda k 'i değerini ifade eder.

Brut Force \rightarrow n tane eleman var, w kapasitesi olanlar
çantası en değerli yapıya k için dahil olacak tüm hedefleri dereriz.
 n adet nesne için 2^n adet alt küme vardır. Hepsine
bakarsak Algoritma karmaşıklığı 2^n olacaktır.

Eğer, w_k (çantaya elemanın ağırlığı $< w$ ise ve k . eleman
kaydedildikten sonra değeri artıyorsa k . eleman çantasıya koyulur.

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

1 → element sayısı 4
 w → çanta kapasitesi 5

Elementler $I(w, v)$ (2,3) (3,4) (4,5) (5,6)

Testime kapasite? \rightarrow testime kapasitesi?

1. 2. 3. 4.

1. 2. 3. 4. nesne

$i \backslash w$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	3	3	3
2	0	0	3	4	4	7
3	0	0	3	4	5	7
4	0	0	3	4	5	6

Bu algoritmin çalışma zamanı $(n \cdot w)$ \rightarrow değeri yüksek olabilir.

Brute force olarak $O(2^n)$

Greedy n'ye göre çözmez ama en

optimum çözümü vermez

